TN 295
.U4

No. 9245

THE LIBRARY OF CONGRESS

THE LIBRARY OF CONGRESS

C219
22

IC 9245

BUREAU OF MINES
INFORMATION CIRCULAR/1990

A User's Manual for MFIRE: A Computer Simulation Program for Mine Ventilation and Fire Modeling

By Xintan Chang, Linneas W. Laage,
and Rudolf E. Greuer

U.S. BUREAU OF MINES
1910 - 1990
THE MINERALS SOURCE

Mission: As the Nation's principal conservation agency, the Department of the Interior has responsibility for most of our nationally-owned public lands and natural and cultural resources. This includes fostering wise use of our land and water resources, protecting our fish and wildlife, preserving the environmental and cultural values of our national parks and historical places, and providing for the enjoyment of life through outdoor recreation. The Department assesses our energy and mineral resources and works to assure that their development is in the best interests of all our people. The Department also promotes the goals of the Take Pride in America campaign by encouraging stewardship and citizen responsibility for the public lands and promoting citizen participation in their care. The Department also has a major responsibility for American Indian reservation communities and for people who live in Island Territories under U.S. Administration.

(United States. Bureau of Mines)

Information Circular/9245

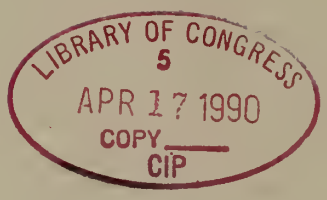
A User's Manual for MFIRE: A Computer Simulation Program for Mine Ventilation and Fire Modeling

By Xintan Chang, Linneas W. Laage,
and Rudolf E. Greuer

UNITED STATES DEPARTMENT OF THE INTERIOR
Manuel Lujan, Jr., Secretary

BUREAU OF MINES
T S Ary, Director

TN295
.U4
no. 9245



Library of Congress Cataloging in Publication Data:

Chang, Xintan.

A user's manual for MFIRE : a computer simulation program for mine ventilation and fire modeling / by Xintan Chang, Linneas W. Laage, and Rudolf E. Greuer.

p. cm. – (Information circular / Bureau of Mines; 9245)

1. Mfire (Computer program). 2. Mine ventilation–Computer simulation. 3. Mine fires–Computer simulation. I. Laage, Linneas W. II. Greuer, Rudolf E. III. Title. IV. Series: Information circular (United States. Bureau of Mines); 9245.

TN295.U4 [TN301] 622 s–dc20 [622'.42'0113] 89-70824
CIP

CONTENTS

	Page
Abstract	1
Introduction	2
MFIRE modifications of MTU-BOM code	2
Program installation and operation	3
Data files	4
Data file management	4
General structure of data file	5
Data lines	5
Instruction lines	5
Comment lines	5
Blank lines	5
Rules for numerical input	5
Detailed description of data input requirements	6
Simulation results	9
Contents of output file	9
Example of simulation	10
Errors	10
Data file errors	10
Program error detection	10
Locating errors	10
Conclusions	12
Appendix A.–MFIRE source code	13
Appendix B.–Example of MFIRE data file and output	141

ILLUSTRATION

1. Waldo mine ventilation network	11

UNIT OF MEASURE ABBREVIATIONS USED IN THIS REPORT

atm	atmosphere	h	hour
Btu/min	British thermal unit per minute	in	inch
cfm	cubic foot per minute	min	minute
°F	degree Fahrenheit	s	second
ft	foot		

A USER'S MANUAL FOR MFIRE: A COMPUTER SIMULATION PROGRAM FOR MINE VENTILATION AND FIRE MODELING

By Xintan Chang,[1] Linneas W. Laage,[2] and Rudolf E. Greuer[3]

ABSTRACT

MFIRE is a computer simulation program that performs normal ventilation network planning calculations and dynamic transient-state simulation of ventilation networks under a variety of conditions. The program is useful for the analysis of ventilation networks under the influence of natural ventilation, fans, fires, or any combination of these. MFIRE simulates a mine's ventilation system and its response to altered ventilation parameters such as the development of new mine workings or changes in ventilation control structures, external influences (varying outside temperatures), and internal influences (fires). Extensive output enables detailed, quantitative analysis of the effects that the alteration will cause to the ventilation system.

The program was developed at Michigan Technological University (MTU), Houghton, MI, with support from the Bureau of Mines. A personal computer (PC) version of the program was completed under contract for the Bureau by MTU. The purpose of this report is to document the installation and use of the PC version of the program MFIRE.

[1]Associate professor of mining engineering, Xian Mining Institute, Xian, Shaanxi, The Peoples Republic of China.
[2]Mining engineer, Twin Cities Research Center, Bureau of Mines, Minneapolis, MN.
[3]Mining engineer, Twin Cities Research Center; professor of mining engineering, Michigan Technological University, Houghton, MI.

INTRODUCTION

MFIRE is useful for the analysis of ventilation networks under thermal and mechanical influence. MFIRE simulates a mine's ventilation system and its response to altered ventilation parameters, external influences, and internal influences. Extensive output enables detailed quantitative analysis of the effects of the proposed alteration on the ventilation system.

MFIRE was written in FORTRAN 77 and compiled with Microsoft[4] FORTRAN for the Intel 8086-8088 central processing units. The source code is listed in appendix A for users who wish to modify the source code for operation on other computers.

MFIRE MODIFICATIONS OF MTU-BOM CODE

The program described in Bureau of Mines Information Circular 8901 has become known as the MTU-BOM code.[5] MFIRE is an evolutionary product of the MTU-BOM code with major improvements and enhancements.[6] For users previously familiar with the MTU-BOM code, the following is a listing of the major improvements and enhancements incorporated by MFIRE.

1. The length of an airway is an input value, LA. The elevation difference of the airway is calculated from the elevation input, Z, in both programs. It is possible for a data file to describe an airway that is shorter than its elevation difference. The MTU-BOM code does not check for this error and uses the values of LA and Z. This does not cause a computational problem, only a simulation problem. MFIRE issues a warning message for this case, and the airway length is made equal to the elevation difference if LA is less than Z.

2. To evaluate rock temperatures (average wall surface temperature of airway before a fire), it is assumed that the air temperature at an airway entrance is equal to the air temperature at the airway starting junction, and that the air temperature at the airway exit is equal to the air temperature at the exit junction.

The MTU-BOM code follows this assumption in all cases. When a rock temperature of less than 0° F is calculated, an average temperature is used to represent the rock temperature of the airway. MFIRE uses the same principle, except the consideration of unlikely rock temperatures. When the exit junction of an airway is the atmosphere (a fixed condition junction), the air temperature at the exit is made equal to the air temperature at the airway beginning. If an unlikely value of wall temperature (<0° F or >150° F) results from calculations, a message is issued and the rock temperature is set equal to the air temperature at the airway exit. The program user can always input rock temperature (TROCK) for an airway to override the calculated value.

3. MFIRE allows use of an optional reference airflow rate to define and limit the heat generation of a fire source. This allows the simulation model to more closely resemble a real fire and to improve simulation convergence. Heat generation corresponds, but is not linearly related to airflow rates reaching the fire. In the simulation, heat generation is calculated from oxygen consumption and is directly related to the variable incoming airflow. Use of this optional reference airflow rate allows the heat generation to vary, but in a damped, user defined manner.

When the airflow rate passing through a fire source is equal to its reference airflow rate, the data items describing the fire source are not modified. If the airflow rate is lower or higher than its reference airflow rate, the data items used to define the fire source are modified according to the following equations:

$$F = 1.0 + (Q - Qr)/Qx$$

where F = the factor to modify the heat generation,

Q = the airflow rate passing through the fire source,

and Qr = the reference airflow rate for the same fire source.

$Qx = Q + Qr$ when $Q < Qr$

or $Qx = Q$ when $Q > Qr$.

Then for fixed heat input fires,

$$HF = H(input)*F;$$

oxygen rich fires,

$$CF = C(input) + [21.0 - C(input)]*[F - 1.0];$$

and fuel rich fires,

$HT = H(\text{per unit } O_2)$ when $Q < Qr$

or $HT = H(\text{per unit } O_2)/F$ when $Q > Qr$

---

[4]Reference to specific products does not imply endorsement by the Bureau of Mines.

[5]Greuer, R. E. Real-Time Precalculation of the Distribution of Combustion Products and Other Contaminants in the Ventilation System of Mines (contract J0285002, MI Technol. Univ.). BuMines OFR 22-82, 1981, 263 pp.; NTIS PB 82-183104.

Greuer, R. E. Study of Mine Fires and Mine Ventilation. Part I. Computer Simulation of Ventilation Systems Under the Influence of Mine Fires (contract S0241032, MI Technol. Univ.). BuMines OFR 115(1)-78, 1977, 165 pp.; NTIS PB 288 231.

Edwards, J. C., and R. E. Greuer. Real-Time Calculation of Product-of-Combustion Spread in a Multilevel Mine. BuMines IC 8901, 1982, 117 pp.

[6]Chang, X. Digital Simulation of Transient Mine Ventilation. Ph.D. Thesis, MI Technol. Univ., Houghton, MI, 1987, 162 pp.

where HF = the heat input with respect to Q,

H(input) = the user-defined value of heat input,

CF = the O_2 concentration behind fire with respect to Q,

C(input) = the user specified value of O_2 concentration behind fire,

HT = the heat generation per unit of O_2 delivery at Q,

and

H(per unit O_2) = the user defined value of heat generation per unit O_2 delivery at Q.

These formulas allow the heat output from a fire source to more realistically change in relation to the varying airflow rate. If the airflow rate is zero, heat output is zero, even for the fixed heat input model. When the airflow rate equals the reference airflow, the data items for heat input are not modified. When the airflow becomes large, the total heat generation rate will increase, but not by more than a factor of 2 times that at the reference condition. This method assumes that heat generation is damped in a manner described by the equations. This optional method provides a more realistic model and allows more user control of the simulation.

4. When sorting airways according to airway types before forming the base system, fixed-quantity airways are placed at the bottom of the sorted list with MFIRE, while arbitrarily mixed fixed-quantity airways and fan branches are placed there with the MTU-BOM code. The reason for the change is that a base system containing a fan is still a mathematically solvable model, while a base system containing a fixed-quantity airway cannot be solved with the normal network balancing techniques. This method reduces the possibility that MFIRE must convert a fixed quantity airway into a normal airway owing to its involvement in the base system.

5. The MTU-BOM code performs fan-curve fitting with Lagrangian interpolation or a least squares method. MFIRE employs the spline method. Usually, the spline method is superior for fan curve fitting for network problems.

6. MFIRE converts the fixed quantity airways into regular airways or fixed pressure fan branches according to the manner that best satisfies the required airflow rates. The MTU-BOM code converts fixed quantity airways into regular airways with positive or negative airflow resistances. This can cause divergence during the network balancing.

7. The difference in natural ventilation pressures in successive iterations is MFIRE's only criterion to determine convergence. The MTU-BOM code uses this criterion, as well as whether an airflow reversal occurred in the last iteration. When an airflow reversal occurs in a sensitive airway with a small pressure loss and no hot fumes entering, this additional criterion is not necessary.

8. MFIRE has a comprehensive data checking procedure. Various messages are used to indicate input error, data inconsistency, or inadequacy. However, a cautious inspection of the data items and good judgement by the user is still required.

9. Both programs assume that a fire is at the entrance of a network branch. The entrance is the end with incoming air. When an airflow reversal occurs, the branch exit and entrance end points swap ends along with the fire location, potentially preventing convergence. This problem could be overcome with the MTU-BOM code by creating a short airway for the fire location. For each contamination source, MFIRE automatically creates a new branch of 0.1 ft in length for the fire location. This prevents of divergence due to an airflow reversal in the fire branch.

10. If MFIRE fails to converge during a simulation, a table showing the causes of the failure is produced. The possible causes are a model without equilibrium conditions or a model requiring more iterations to satisfy the convergence criteria. For example, a fire located in a descensionly ventilated airway may have no physically or mathematically stable equilibrium state. The table of the intermediate results with the mean air temperature and airflow rates in airways will show a definite cyclic pattern that will not converge with more iterations. If the convergence process is slower than normal, but the convergence criterion has been approached consistently, more iterations may complete the convergence.

11. Transient-state modeling (with user defined intervals) of the effect of mine fires on a mine ventilation system and propagation of contaminants is accomplished with MFIRE.

12. A user defined timetable to change simulation conditions during the transient-state simulation stage makes the program more flexible for modeling and simulation.

PROGRAM INSTALLATION AND OPERATION

The program MFIRE consists of four files: MFIRE0.EXE, MFIRE1.EXE, MFIRE2.EXE, and MFIRE.BAT. The batch file, MFIRE.BAT, controls the operation and cooperation of the three executable program files and the output. Networks of up to 500 airways and 350 junctions can be modeled in the PC version of MFIRE. Larger data sets can be accommodated by increasing array dimensions and compiling the program again.

Input for MFIRE is in the form of a data file. The data file contains lines of numerical data describing the physical mine, the ventilation parameters present, the fans, and the proposed thermal or mechanical disturbances.

Output from MFIRE is sent to the file MFIRE.OUT, which is created in the directory where the MFIRE program resides. The output can have up to four levels of increasing detail. Subsequent executions of MFIRE will overwrite the old MFIRE.OUT file with the new MFIRE.OUT file. If archiving of MFIRE.OUT files is desired, they can be renamed with the (MS-DOS) RENAME command to prevent overwriting. The MFIRE.OUT file can be reviewed and/or printed with text editing software.

The equipment and additional software necessary to use MFIRE includes:

1. IBM PC or compatible computer with MS or PC DOS 2.0 or later version.
2. Hard disk drive (not required, but highly desirable).
3. Intel 8087 or compatible math-coprocessor (not required, but highly desirable).
4. At least 256K random access memory (RAM).
5. Text editor or suitable word processor.

To install the program, the user should make a directory on the hard disk drive. The contents of the program disk are then copied into that directory. To start MFIRE, the user types "MFIRE" and follows the instructions for user responses and the location of the data file.

MFIRE can be run from a floppy disk; however, temporary files and the output file will then be placed on the same floppy disk. Running MFIRE from a floppy disk may cause problems due to storage limitations. To run MFIRE from a floppy disk, insert the disk, change the default disk drive to the drive containing the disk and type "MFIRE."

Two screens of information about the program are presented followed by a prompt for the user to identify the location of the data file. Depending upon the user defined selection of program features, MFIRE will sequentially perform:

1. General network balancing.
2. Network balancing considering temperatures, heat exchange with rock, and natural ventilation.
3. Dynamic, transient-state modeling with user selected dynamic fire growth, and user selected time increments for updating the ventilation changes and spread of contaminants.
4. Calculation of new ventilation equilibrium due to contaminant (fire) presence, and distribution of the contaminants such as products of combustion (POC) and methane under the altered steady-state conditions.

DATA FILES

Data or input files are used to describe the mine's ventilation system and physical characteristics necessary for ventilation calculations as well as the physical or mechanical disturbance to be modeled and simulated. Data files are organized into lines of categories or groups of information. Each line contains multiple sets of information or data items. Data lines are read by the program, and the data items provide information to the program variables. The data file is in American Standard Code for Information Interchange (ASCII) format, sometimes called a DOS text format. Creation or modification of a data file is accomplished with text editor software. Most word processors are also capable of making the required ASCII type files. A sample data file and its associated output are included in appendix B.

When selecting software for editing the data files, the user should be aware that the software will also be used for viewing the output of the simulation. The output file can be quite large (several megabytes), if a lengthy simulation is performed. The output can be sent to a printer for a hardcopy listing; however, the user should also be aware that most printers currently used with PC's print at rates of 100 to 300 characters per second and large output files may take several hours to print. Viewing the output with a text editor can save much time and paper.

DATA FILE MANAGEMENT

The data files describing the event and mine to be simulated can reside either on a floppy disk or in a directory on the hard disk drive. When the program starts, the user is prompted for the data file location and name. The user response is in the form of a standard DOS path and file name. For example, if the data file is on a floppy disk located in drive B and the disk has a directory MYMINE with the data file MYDATA.DAT in that directory, then, at the prompt:

NAME YOUR INPUT FILE BY
"D:\PATH\FNAME1.EXT"

the user would type

B:\MYMINE\MYDATA.DAT

followed by a carriage return or the ENTER key.

Any legal DOS path, file name, and extension can be used; however, the user may find it convenient to create a directory for data or input files and to use DAT or some other meaningful extension to identify these data files.

GENERAL STRUCTURE OF DATA FILE

The input data file for MFIRE may have four types of line records: data lines, instruction lines, comment lines, and blank lines. Only the data lines are required.

Data Lines

A line whose first non-blank character is either:

0 1 2 3 4 5 6 7 8 9 . , - +

is a data line that provides numerical input values to data items to define the mine's ventilation, the physical properties, and the event to be simulated.

If a data line contains a symbol that is not from the above list, an error message will be given. If a line contains three or more characters that are not from the list, the line will be regarded as a comment line and printed in the output file.

Altogether, data lines comprise the following data groups:

Comments (optional).
Control line number 1.
Airway lines.
Junction lines (optional).
Fan lines (optional).
Fire source lines (optional).
Control line number 2 (optional).
Average-value lines (optional).
Timetable lines defining condition changes (optional).

Instruction Lines

A line whose first non-blank character is a dollar sign ($) is an instruction line to help the user setup and organize data files. Users can add or delete instruction lines. Instruction lines do not appear in the output file and do not interfere with the program execution.

A data file must be setup according to the input requirements. Because errors tend to occur in the preparation of data file, a template file, INPUT.FMT was created. This file consists of instruction lines detailing the data line groups in the required order and the location of each data item on each line as well as the meaning of each data item required. This file serves to remind and prompt the user for data lines and items at the proper locations. Appendix B shows the INPUT.FMT file filled with sample data.

To use the template, simply copy the file INPUT.FMT to the data disk space, and add data into the data file by adding lines at proper places and following the instructions given in the INPUT.FMT file. The user can then use the DOS RENAME command to identify uniquely the new data file.

Comment Lines

A line whose first non-blank character is not one of the above specified input data characters, or which contains more than 3 characters that are not data input characters, is a comment line. Comment lines are created by the user to record notes about the data file. Comment lines will appear in the output list to identify the data file producing the output and/or to provide notes to the user concerning the data file. The comment lines within the airway data paragraph will be automatically centered. Because the width of an output line is 72 columns, a comment line should have a non-blank width (spaces between the first and last non-blank characters) of 72 characters or less.

A comment line can be put anywhere in the data file, but only the lines located before control line number 1 and the ones within the airway data lines group will be shown in the output file. The maximum number of total comment lines is 50.

Blank Lines

A line containing only blanks is a blank line which has no effect on the program.

There is no limit on the number of instruction lines and blank lines, and they may be placed anywhere in the data file.

RULES FOR NUMERICAL INPUT

MFIRE was written in FORTRAN and compiled with the Microsoft FORTRAN Compiler into executable files. Free format is used for the MFIRE data files.

1. Distinction between an integer and a real number is not required. Integer 5 can be 5 or 5.0 in the data file. Real number 3.0 can be 3 or 3.0 in the data file.
2. A comma and a blank space can be used interchangeably as delimiters to separate input values. Two or more blank spaces adjacent to one another act as one blank for input separation. Blank spaces adjacent to a comma have no effect on separating input values.
3. Zero values can be omitted, but the related delimiters (commas) cannot, except the case where the remaining characters on the line are zero.

For example, if input is needed for the following data items:

N,R,S,I,T,L,P,M

where N=5, R=10, S=7.52, I=0, T=-4, L=0, P=0, and M=0,

then the following input lines would have the same effect:

```
5,10.0,7.52,0,-4.0,0,0.0,0
5 10 7.52 0 -4 0 0 0
5 10 7.52 , , -4
5,10,7.52,,-4
5 10 7.52,,-4
```

4. Comment lines can be put anywhere in an input file. Comment lines will be printed out in output files, only if the comments are located before the control line number 1 or within the group of airway data lines.

5. Instruction lines and blank lines can be put anywhere in a data file. They will not affect the program run.

DETAILED DESCRIPTION OF DATA INPUT REQUIREMENTS

Optional data groups can be omitted only if indicated by the input value of the controlling data item. Comments and numerical data must be in the following order:

1. Comments (optional).

Comments here and within the airway data group will be shown in the output file. This feature provides a section for user notes and comments concerning a data file.

2. Control line number 1 consists of the following variables:

NB,NFNUM,INFLOW,NVPN,NETW,TR,MADJ,ITN, NTEMP,TINC,SPAN,IOUT,TOUT,CONCT

where

NB is the number of airways in the network. If the value of NB >1, its value will be used to determine the number of input airway lines. Otherwise (when NB=1 or NB<1), the airway lines group must end with an airway line that has an airway identification number (ID) >9990. In this special case all other items in the finishing airway line can be omitted. Maximum NB = 500.

NFNUM is the number of fans which have characteristic curve input data. Maximum NFNUM = 10.

INFLOW is the total number of contamination sources in the system. Maximum INFLOW = 10.

NVPN is a flag that allows simulation of natural ventilation. If there are junction cards to be read in, NVPN must be ≤0. The junction lines must be omitted if NVPN <0. When the junction cards are omitted, only the network simulation will be performed.

NETW is a flag. When NETW = 1, only the network simulation will be performed. The output will consist of conventional network calculations without consideration of contaminant sources.

TR is the reference or air average temperature in the mine in degrees Fahrenheit. If the input value of TR is <10° F or >110° F, the default value of 75° F will be used.

MADJ is the maximum number of iterations in the temperature section. If its input is <10 or >80, the default value of 10 will be used.

ITN is the maximum number of iterations in the dynamic simulation section. If its input is <10 or >80, the default value of 10 will be used.

NTEMP is a flag. When NTEMP = 1, the simulation will terminate after the steady-state simulation.

TINC is the time increment for the dynamic simulation section in seconds. If the input value is ≤0, default value of 15 s will be used.

SPAN is the time span for the dynamic simulation of the proposed event in minutes.

IOUT is a marker to define the level of detail of the output file, where: 0=brief; 1=normal; -1=detailed; and -2 = most detailed.

TOUT is the interval in minutes for output of the simulation. The default value is 1 min. Whenever significant changes occur in the system, output will automatically be provided regardless of the TOUT value.

CONCT is a marker for the transient concentration simulation. When CONCT=1, the transient concentration simulation is performed with the conditions of constant airflow and temperature distribution. When CONCT=0, varying airflow and temperature distributions are considered. CONCT should normally be set to 0.

3. Airway lines. Each line describes a single branch in the network to modeled. Each line consists of the following variables:

NO,JS,JF,NWTYP,R,Q,KF,LA,A,O,HA,HK,CH4V, CH4PA,TROCK

where

NO is the airway ID. NO must be between 1 and 9990.

JS is the starting junction ID number of the airway. JS must be between 1 and 999.

JF is the ending junction ID number of the airway. JF must be between 1 and 999.

NWTYP is the marker for airway type. Values of NWTYP can be as follows: 1 = fan branch (with zero airflow resistance), 0 = normal airway, and -1 = fixed quantity airway (regulator). A regulator will be converted to a normal airway or fan branch after the network balancing is performed depending upon the method by which the user-specified airflow requirement can be satisfied. During the simulation, a fire source branch will automatically be given NWTYP value of 1.

R is the resistance of a normal airway in $1.0*E^{-10}$ inches water gage divided by cubic feet per minute squared (in wg/$(cfm)^2$). When the input value of R is ≤0, R will be calculated from KF, LA, A, and O. If these input data items are not complete, an error message will be issued. When the input value of R is >0, it will override the value calculated from KF, LA, A, and O. For a fan branch (NWTYP=1), R is the fan pressure in inches water gage.

A fan without input for its characteristic curve will be regarded as a fixed-pressure fan and the value of R will be used to define the fan pressure.

Q is the volumetric airflow rate in the branch in cfm.

KF is the friction factor in $1.0*E^{-10}$ pounds mass times minutes squared divided by feet to the fourth power ($lb*min^2/ft^4$). The default value = 100.

LA is the length of the airway in feet (ft). The default value = 500.

A is the cross-sectional area of airway in square feet (ft^2). The default value = 100.0.

O is the perimeter of airway in feet (ft). The default value = 40.0.

HA is the thermal diffusivity of rock in square feet divided by hours (ft^2/h). The default value = 0.1.

HK is the thermal conductivity of rock in British thermal units divided by hours times feet times degrees Fahrenheit (Btu/h*ft*°F). The default value = 3.0.

CH4V is the methane emission rate for the whole airway in cubic feet divided by minutes (ft^3/min) under the reference conditions (1 atm and TR). If CH4V is ≤0, the methane emission rate in the airway will be calculated from CH4PA.

CH4PA is the methane emission rate per unit surface area of the airway in cubic feet divided by minutes times square feet ($ft^3/min*ft^2$) under the reference conditions. If CH4PA is ≤0, the methane emission rate will be calculated from the methane concentration in junctions (CH4C). If both CH4V and CH4PA are 0, methane emissions are not considered.

TROCK is the rock temperature in degrees Fahrenheit of the airway (quasi-equilibrium wall surface temperature before the fire started). It is normally calculated in the program run and does not require input from the user. If the user has better data available, it may be input here to override the calculated value.

The acceptable range of user specified rock temperature is from 0° to 150° F.

If data items are not available or only a preliminary simulation is required, the default values may be used. However, greater modeling accuracy requires more accurate data for the various input variables. The potential effect of approximating these data items must be considered when using the respective default values. When the default values are wanted, input zero for the item.

4. Junction lines section 1 are used to describe conditions and locations of the junctions. When NVPN is negative or zero, each junction referenced in the airway data section must have a one line input. Otherwise, the junction lines must be omitted.

Junction lines section 1 consist of four variables:

JNO,T,Z,CH4C

where

JNO is the junction ID number. It must be between 1 and 999.

T is the junction temperature, °F.

Z is the junction elevation in feet above a datum level.

CH4C is the methane concentration in percent in the junction.

5. Junction lines section 2 are employed when the ventilation system has multiple atmospheric junctions that have different elevations. The junctions should be connected with very low resistance and high-airflow branches, which represent the atmospheric connection between the junctions. Junction lines section 2 consist of three variables:

JAN,JA,JATP

where

JAN is the number of atmospheric junctions. JAN is on the first line of the data group.

The remainder of the data group consists of lines, each line describing an atmospheric junction with: JA,JATP

where

JA is the ID number of the atmospheric junction.

JATP is a marker for the type of the atmospheric junction. 0 is the value for the starting junction for the simulation and 1 is the value for other atmosphere junctions, except starting junction.

6. Fan line number 1 consists of two variables:

NOF,MPTS

where

NOF is the ID number of the airway in which the fan is located.

MPTS is the number of data pairs describing the fan characteristics. Maximum MPTS = 10.

7. Fan line number 2 consists of data pairs describing the fan characteristic curve in the form:

QF,PF, QF,PF, QF,PF, QF,PF, QF,PF

where

QF is the airflow through the fan in cubic feet per minute (cfm) and PF is the fan pressure in inches water gage corresponding to QF.

For each data point, a pair of QF and PF data must be specified. Up to 5 pairs of QF and PF data points may be

on a line. A maximum of 10 data points or two lines may be used to describe the fan characteristics of each fan. The line order of the input points of the fan curve must be in ascending numerical order from small QF to large QF.

Fan lines numbers 1 and 2 describe one fan. Additional fans are described by adding another set of fan lines numbers 1 and 2. The number of fans described here must equal NFNUM on control line number 1. If NFNUM is ≤0, fan lines 1 and 2 must be omitted.

8. Contamination (fire source) lines consist of up to nine variables. Data item INFLOW in control line number 1 indicates the number of contamination sources that are present in the ventilation system. Each of the contamination sources requires a one line data input. If the input value of INFLOW in control line number 1 is ≤0, contamination lines must be omitted.

Through appropriate selection of values for variables CONT, CONC, HEAT, O2MIN, SMPO2, and HTPO2, the user can simulate a fixed heat input fire, an oxygen-rich fire, or a fuel-rich fire.[7]

Contamination lines may include the following data items:

NCENT,CONT,CONC,HEAT,O2MIN,SMPO2,
HTPO2,QCENT,TPR

where

NCENT is the ID number of the airway with the fire source located at the entrance of the incoming airflow.

CONT is the contaminated gas inflow from the contamination source in cubic feet per minute.

CONC is the concentration of fumes in the contaminated gas, CONT, in percent.

HEAT is the heat influx at the fire source in British thermal units per minute (Btu/min).

O2MIN is the oxygen concentration leaving the fire source in percent.

SMPO2 is the fume production in cubic feet for each cubic foot of oxygen reaching the fire source.

HTPO2 is the heat generation in British thermal unit for each cubic foot of oxygen reaching the fire source.

QCENT is the reference airflow in cubic feet at which the above parameters are defined. When the airflow is less than or greater than QCENT, the heat generation from the fire source will decrease or increase accordingly, but in a damped fashion. If QCENT is <10 cfm the damping is not performed. For details refer to the introduction.

TPR is the leading period in minutes for an oxygen rich fire to reach full strength. Heat generation is calculated from available oxygen. Increasing the airflow to a fire will increase heat output, however the heat output will lag the increased airflow. The quantity of lag will vary with the fuel type and is user defined with TPR. The effect is linear over the time TPR.

CONT, CONC, and HEAT define a fixed heat input fire source. O2MIN defines an oxygen rich fire. SMPO2 and HTPO2 define a fuel rich fire. When they all have nonzero input values, the oxygen rich model overrides fixed heat input fire model, while the fuel rich fire model overrides the oxygen rich model.

9. Control line number 2 consists of the following variables:

NAV,JSTART,TSTART,TIME,CRITSM,CRITGS,
CRITHT,WRNPR,WRNSM,WRNGS,WRNHT

where

NAV is a marker. If NAV is ≤1, an average value card is present in the data file to supply required values, which may have been omitted in the data file.

JSTART is the ID number of starting junction in the ventilation system. At the junction of JSTART (usually the atmosphere), conditions never change.

TSTART is the air temperature at the starting junction in degrees Fahrenheit.

TIME is the time span to assume quasi-equilibrium in hours. Default 5 h.

CRITSM is the accuracy criterion for fume calculation in percent. The default value = 0.005.

CRITGS is the accuracy criterion for methane calculation in percent. The default value = 0.01.

CRITHT is the accuracy criterion for temperature calculation in degrees Fahrenheit. The default value = 0.1.

WRNPR is the pressure drop criterion in inches water gage for issuing a warning message. The default value = 0.01.

WRNSM is the fume concentration criterion in percent for issuing a warning message. The default value = 0.05.

WRNGS is the methane concentration criterion in percent for issuing a warning message. The default value = 1.0.

WRNHT is the high temperature criterion in degrees Fahrenheit for issuing a warning message. The default value = 100.0° F.

The last seven data items serve as calculation accuracy criteria or warning message issuing criteria. Usually, the available default values will be suitable. To activate the respective default values, place a zero at that location in the data file.

10. The average value line consists of a set of average values for data items that are required for the program to run. The average value line supplies these data items if they cannot be obtained from the data file or default values. These items are used to supply unknown or omitted

[7]Greuer, R. E. Influence of Mine Fires on the Ventilation of Underground Mines (contract S0122095, MI Technol. Univ.). BuMines OFR 73-074, 1973, 173 pp.; NTIS PB 225-834.

omitted data. The presence of an average value line is indicated by a value ≥1 for the variable NAV in control line number 2. These items include:

HAAVR,HKAVR,KFAVR,LAAVR,AAVR,OAVR

where

HAAVR is the average thermal diffusivity of rock in square feet per hour. The default value = 0.1.

HKAVR is the average thermal conductivity of rock in Btu/h*ft*F. The default value = 3.0.

KFAVR is the average friction factor in $1.0*E^{-10}$ $lb*min^2/ft^4$. The default value = 100.

LAAVR is the average length of an airway in feet. The default value = 500.

AAVR is the average cross-sectional area of an airway in square feet. The default value = 100.0.

OAVR is the average perimeter of an airway in feet. The default value = 40.0.

To select a default value, use zero for the respective data item. If the entire set of default values are to be used, set NAV=0 on control line number 2 and omit the average value line.

11. Timetable lines for describing condition changes during the dynamic simulation can consist of seven types of condition changes. The general input scheme for each line is: TS, ICODE, (parameters). TS indicates the starting time of the condition change, and ICODE defines the nature of the condition change. Timetable lines, ICODE's, and the parameters are defined as:

1) TS,1,NBR,R

At TS minutes after time zero, branch NBR becomes an ordinary airway with its resistance equal to R.

2) TS,2,NBR,INDEX,QF1,PF1,QF2,PF2,QF3, PF3,QF4,PF4,QF5,PF5

At TS minutes after time zero, branch NBR becomes a fan branch with its characteristics defined by QF1 to PF5. INDEX indicates the number of input data points. If INDEX equals 1, the fan is a fixed pressure fan.

3) TS,3,NBR,CONT,CONC,HEAT,O2MIN, SMPO2,HTPO2,QCENT,TPR

At TS minutes after time zero, branch NBR become a fire source branch with its characteristics defined by CONT to TPR. These parameters retain the same meaning as in data group 8.

4) TS,4,NBR

The previously started fire in branch NBR is extinguished at TS minutes after time zero.

5) TS,5,TINC

The time increment in dynamic simulation is changed to TINC seconds at TS minutes after time zero.

6) TS,6,TOUT

The interval for output is changed to TOUT minutes at TS minutes after time zero.

7) TS,7,JDPP

At time TS detailed output results will be shown for airways on the immediate downstream side of junction JDPP.

Each of the condition changes requires a one line input. There is no limit for the number of lines. The order of lines must be from small TS to large TS. This last line of the table must be a number (TS without an associated ICODE) larger than the simulation time span (SPAN on control line number 1). When NETW or NTEMP on control line number 1 is >0, this timetable can be omitted. Otherwise, it must exist, but may contain only the ending line.

SIMULATION RESULTS

All output is saved in the file MFIRE.OUT in the directory and on the disk containing the program. The file can be inspected with a text processor and sent to a printer, if needed.

CONTENTS OF OUTPUT FILE

The comment lines, reference temperature, air density, a table of the units of measure for the results, and the average values are shown at the beginning of MFIRE.OUT.

All input data will be shown in the output file if IOUT on Control line number 1 is ≤-1. For each contamination source, a new branch will automatically be setup. The new airway and junction ID numbers will not interfere with the original numbering scheme of the data file.

Next are the results of the selected simulation(s) in the following order: network balancing, network balancing considering temperatures, transient-state modeling, and altered equilibrium due to fire and distribution of the contaminants.

Depending upon the contents of the data file, and the simulation selected, attention, warning, and error message will be shown at locations where calculations resulted in unusual conditions.

EXAMPLE OF SIMULATION

Appendix B contains an example of a simulation for the Waldo Mine, which exercises most of MFIRE's options. The Waldo Mine is an underground mine owned by Asarco and leased to New Mexico Institute of Mining and Technology for research purposes.

Figure 1 shows a simplified network of the Waldo Mine. The Waldo mine has six atmospheric junctions. Level 3 has adits with entrance junction numbers 27 and 34. Level 3 also has surface leakages modeled by airways 39, 47, and 40 to junction 1. Junction 3 is the surface entry of a shaft with a downcast fan. Junction 4 is the surface entry for level 9 adit. Junction 33 is at the surface above level 3. These surface junctions are connected by low resistance, high flow rate airways 48, 41, 42, 43, and 44. Junction 1 appears in three physical locations and junction 27 appears in two physical locations on the network. These six junctions are at different elevations.

The network characteristics are listed in the input file. A fire was simulated in airway 4 for 2 min with output every minute. The fire provided 50,000 Btu/min of heat and 200 cfm of pure products of combustion (POC) to the network. Examination of the output file shows the results of the network balancing and the transient-state fire simulation. Information is provided on the airway reversals, POC location, critical levels of POC, temperature, and pressure loss, as well as ventilation parameters for each interval.

ERRORS

The usual run time errors occur when illegal mathematical operations are attempted by MFIRE. For example, if the user enters a negative value for the input variable HEAT to model a heat sink or cooling station, a run time error will occur because an exponential function is employed to better model the changing temperature distribution in the airways.

DATA FILE ERRORS

Two types of data file errors can occur.

1. An error may occur because the input data file itself is invalid. This may be caused by:

The data file cannot be found due to incorrect spelling, drive, or path specification. Check the spelling and path at the prompt for location of the data file.

The data file contains non-ASCII symbols. Check operation of the text processor to insure that only ASCII characters are produced. Delete all non-ASCII and/or all out of place characters.

2. Data are accepted by the program, but are incorrect for the data types or ranges or the combination causes a data inconsistency. This may be due to:

An invalid character is located where a numerical value is required.

The input data value is outside the reasonable range.

The input data is insufficient to build the model for simulation.

The input data exceeds the capacity of the program.

PROGRAM ERROR DETECTION

When the program is running, tracing messages are given on the screen indicating the group and lines of data that have been read and the section of the program that is executing. Program termination indicates that problems are caused by the subsequent (not read) data file section.

For each detected error, descriptive message starting with * ERROR * is issued and the program terminates.

When a possibly wrong value and/or its calculated intermediate result is detected, a descriptive message starting with * WARNING * is issued and the program continues.

If a modification of input value or calculated intermediate result is automatically made by the program, descriptive message starting with * ATTENTION * is issued and the program continues.

LOCATING ERRORS

If a message indicates that the error condition occurred due to the data input file, locate and correct the data item.

If no error message is issued, but the simulation results are not as expected, carefully check the list of input data shown in the output file.

WARNING messages and ATTENTION messages may reveal an error in the input data even though the program continues to operate. Insure that the default values or modifications adopted by the program run are acceptable.

Anomalistic data, though not incorrect or impossible, may lead to an unusual simulation. For example, making a fan unusually powerful may prevent a normal strength fire from affecting the airflow distribution in the system. Much care must be employed before considering an extreme case.

If a message indicates that the accuracy criterion in the temperature section of the program was not satisfied, a table detailing the current parameters of airways which have large fluctuation in the average air temperature from iteration to iteration is produced. If the pattern of the parameters is clearly cyclical, while the variation of natural ventilation pressure is stable or cyclical from iteration to iteration, then, most likely there is no single equilibrium stage for the event. If the variation of natural ventilation pressure decreases steadily though slowly, increasing the value of data item variables MADJ or ITN (on control line number 1) may allow convergence.

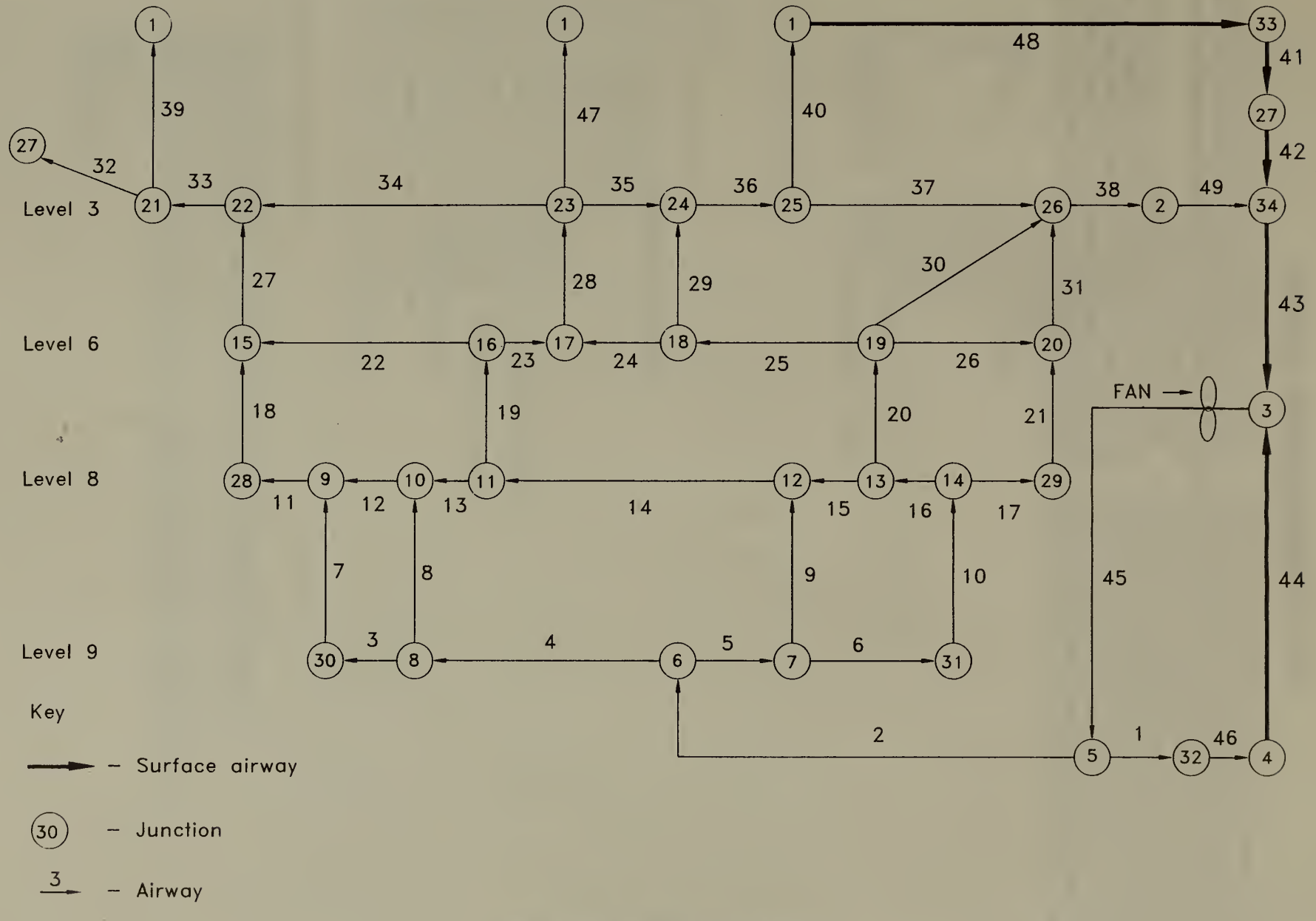


Figure 1.–Waldo mine ventilation network.

CONCLUSIONS

The PC version of the program MFIRE contains many improvements and enhancements previously unavailable for the modeling of mine ventilation networks and the interaction of fires with mine ventilation. Transient-state simulation with dynamic modifications of the ventilation system and mine, the fire, and resulting POC transport and ventilation alterations are accommodated with MFIRE.

MFIRE is more flexible and provides more error checking, and more informational messages about problems with input files than its predecessor, the MTU-BOM code. The desirable features of the MTU-BOM code such as calculation and identification of recirculation, the air to rock heat exchange, and the basing of calculations on mass flow rates have been retained.

MFIRE is useful for the planning and analysis of ventilation networks, simulation for emergency planning, and the evaluation of fire detector layout. Inquiries relating to this program should be directed to the Bureau of Mines, Twin Cities Research Center, 5629 Minnehaha Avenue South, Minneapolis, MN 55417.

APPENDIX A.–MFIRE SOURCE CODE

UNIT OF MEASURE ABBREVIATIONS USED IN MFIRE

FT.	foot
FT3/MIN	cubic foot per minute
LBM*MIN2/FT4	pound mass times minutes squared over feet to the fourth
FT2/HR	square feet per hour
BTU/HR*FT*F	British thermal unit divided by hours times fimes feet times degree Fahrenheit
IN.W.G.	inches water gage
FT3/MIN	cubic feet per minute
FT2	square feet
FT3	cubic feet
LBM;	pound mass
F	Fahrenheit
IN.W.G./CFM2	inches water gage per cubic foot per minute squared
PERCENTAGE	percent
BTU/MIN	British thermal unit per minute

MFIRE.BAT PROGRAM

```
cls
echo off
echo MFIRE activated...
mfire0
echo Wait a moment for loading the program...
if EXIST zheng.chg mfire1
if EXIST zheng.chg mfire2
if EXIST zheng.chg mfire1
if EXIST zheng.chg erase zheng.chg
if EXIST liang.chg erase liang.chg
if EXIST yunzhi.fan copy yaodong.xin+yunzhi.fan yaodong.xin
if EXIST yunzhi.fan erase yunzhi.fan
if EXIST xintan.chg copy yaodong.xin+xintan.chg yaodong.xin
if EXIST xintan.chg erase xintan.chg
if EXIST yaodong.xin copy yaodong.xin mfire.out
if EXIST yaodong.xin erase yaodong.xin
if EXIST net erase net
echo Your output is saved in file MFIRE.OUT
```

MFIRE0.FOR PROGRAM

```
      CHARACTER*50 A
      OPEN (10,FILE='NET',STATUS='UNKNOWN',FORM='UNFORMATTED')
      NETWW=0
      WRITE (10) NETWW
      OPEN (21,FILE='ZHENG.CHG',STATUS='UNKNOWN',FORM='UNFORMATTED')
      A='                                                  '
      B=0.0
      WRITE (21) A
      WRITE (21) A,B
```

```
      STOP
      END


MFIRE1.FOR PROGRAM

C
C     TRANSIENT-STATE SIMULATION PROGRAM FOR MINE VENTILATION
C                       UNDER THE INFLUENCE OF FIRES
C
C                              VERSION OF JULY 1989 V 1.20
C                                    (PART 1)
C
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
$INCLUDE:'CMMN.DAT'
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     PARAMETER  ( NMX=500, NMY=350, NMZ=50, IMX=10, IMY=10, IMZ=10,
C    .                   LMX=10000 )
C
C     COMMON  /CONTRL/ NB,NJ,NFNUM,NVPN,NETW,NTEMP,MADJ,ITN,DR,TR,TINC,
C    .                         SPAN,IOUT,TOUT,CONCT
C
C     NB: NUMBER OF AIRWAYS IN NETWORK.
C     NJ: NUMBER OF JUNCTIONS IN NETWORK.
C     NFNUM: NUMBER OF FAN CHARACTERISTICS.
C     NVPN: MARKER FOR PRESENCE OF JUNCTION CARDS, 0: PRESENT.
C     NETW: MARKER, 1: ONLY THE NETWORK PART WILL BE PERFORMED.
C     NTEMP: MARKER, 1: ONLY UP TO THE TEMP. PART WILL BE PERFORMED.
C     MADJ: MAX. NO. OF ITERATION IN TEMP. PART.
C     ITN: MAX. NO. OF ITERATION IN NON-STEADY STATE SIMULATION.
C     DR: REFERENCE DENSITY OF AIR, LBM/FT3.
C     TR: REFERENCE TEMPERATURE OF AIR, DEG. F.
C     TINC: TIME INCREMENT IN SIMULATION, SEC..
C     SPAN: TIME SPAN OF SIMULATION, MIN..
C     IOUT: OUTPUT. 0: BRIEF; 1: NORMAL; -1: DETAIL; -2: MORE DETAILED.
C     TOUT: TIME INTERVAL FOR OUTPUT, MIN.
C     CONCT: MARKER. 1: SIMULATION FOR TRANSIENT CONCENTRATION SIMULA-
C                           TION ONLY
C
C     COMMON  /CTRL/ NAV,MAXJ,INFLOW,CRITSM,CRITGS,CRITHT
C
C     NAV: MARKER FOR PRESENCE OF AVE. VALUE CARDS.
C     MAXJ: HIGHEST JUNCTION NUMBER.
C     INFLOW: NUMBER OF CONTAMINATION CARDS.
C     CRITSM: ACCURACY OF FUME CALCULATION.
C     CRITGS: ACCURACY OF METHANE CALCULATION.
C     CRITHT: ACCURACY OF TEMPERATURE CALCULATION.
C
C     COMMON  /NTWK/ NO(NMX),JS(NMX),JF(NMX),NWTYP(NMX),R(NMX),Q(NMX),
C    .                   P(NMX),KF(NMX),LA(NMX),A(NMX),O(NMX),RSTD(NMX),
C    .                   DZRD(NMX),FRNVP(NMX),NREV(NMX),RDCH4(NMX),
C    .                   RDPROP(NMX),TRD(NMX),TJS(NMX),RDOP(NMX),RCH4(NMX),
C    .                   FFRNVP(NMX),RA(NMX),NNREV(NMX)
C
```

```
C    NO(NMX): AIRWAY NUMBER.
C    JS(NMX): JUNCTION NUMBER OF AIRWAY BEGINNING.
C    JF(NMX): JUNCTION NUMBER OF AIRWAY END.
C    NWTYP(NMX): AIRWAY TYPE.
C    R(NMX): RESISTANCE OF AIRWAY 10**10 IN. W.G./(CFM)**2.
C    Q(NMX): BALANCED AIRFLOW RATE, ASSUMING EQUILIBRIA, CFM
C    P(NMX): CALCULATED PRESSURE LOSS IN IN. W.G..
C    KF(NMX): FRICTION FACTOR.
C    LA(NMX): AIRWAY LENGTH IN FT.
C    A(NMX): AIRWAY CROSS SECTIONAL AREA IN FT**2.
C    O(NMX): AIRWAY PERIMETER IN FT.
C    RSTD(NMX): RESISTANCE FACTOR BASED ON TR.
C    DZRD(NMX): ELEVATION CHANGE IN AIRWAY IN FT.
C    FRNVP(NMX): PRODUCT TMRD*DZRD.
C    NREV(NMX): NUMBER OF AIRFLOW REVERSALS IN AN INTERVAL.
C    RDCH4(NMX): METHANE CONCENTRATION AT ROADWAY ENDS.
C    RDPROP(NMX): CONTAMINANT CONCENTRATION AT ROADWAY ENDS.
C    TRD(NMX): TEMPERATURE AT ROADWAY ENDS.
C    TJS(NMX): TEMPERATURE AT ROADWAY BEGINNING.
C    RDOP(NMX): BACKUP ARRAY FOR RDPROP.
C    RCH4(NMX): BACKUP ARRAY FOR RDCH4.
C    FFRNVP(NMX): DZRD/TMRD.
C    RA(NMX): BACKUP ARRAY FOR R.
C    NNREV(NMX): NUMBER OF AIRFLOW REVERSALS SINCE TIME ZERO..
C
C    COMMON  /FAN/ NOF(IMX),NFREG(IMX),RGRAD(IMX),NFCW(IMX),MPTS(IMX),
C   .                  QF(IMX,IMY),PF(IMX,IMY),NSKP(IMX),NEGQ(IMX),JSB(NMX)
C
C    NOF(IMX): AIRWAY NUMBER OF FAN.
C    NFREG(IMX): LIST OF FANS WITH CHARACTERISTICS.
C    RGRAD(IMX): SLOPE OF FAN CHARACTERISTIC.
C    NFCW(IMX): LIST OF FANS WHOSE CHAR. EXCEEDED.
C    MPTS(IMX): NUMBER OF POINTS DEFINING FAN CHAR.
C    QF(IMX,IMY): AIRFLOW AT GIVEN POINT OF FAN CHAR.
C    PF(IMX,IMY): PRESSURE AT GIVEN POINT OF FAN CHAR.
C    NSKP(IMX): MARKER INDICATING PERFORMANCE OF SPLINE.
C    NEGQ(IMX): MARKER INDICATING AIRFLOW REVERSAL IN FAN BRANCH.
C    JSB(NMX): BACKUP ARRAY FOR JS.
C
C    COMMON   /FC/  FC(IMY,IMY),DK(IMY),FK(IMY),FKQ(IMX,IMY,4)
C
C
C
C    COMMON  /MESH/ MNO,MEND(NMY),MSL(LMX),FNVP(NMY),RQ(NMX),INU(NMX),
C   .             KNO(NMX),KJS(NMX),KJF(NMX)
C
C    MNO: NUMBER OF MESHES IN NETWORK.
C    MEND(NMY): LIST OF MESH ENDS IN MSL-LIST.
C    MSL(LMX): LIST OF ALL INDEPENDENT MESHES.
C    FNVP(NMY): LIST OF NVP IN MESHES.
C    RQ(NMX): AUXILIARY LIST OF R*Q.
C    INU(NMX): AUXILIARY LIST FOR FORMING BASE SYSTEM.
C    KNO(NMX): LIST OF AIRWAYS IN BASE SYSTEM.
C    KJS(NMX): LIST OF JUNCTIONS IN BASE SYSTEM.
```

```
C    KJF(NMX): LIST OF JUNCTIONS IN BASE SYSTEM.
C
C    COMMON  /JUNCT/ JNO(NMY),T(NMY),Z(NMY),CH4C(NMY),JNOL(NMY),
C   .                PROP(NMY),PRCH4(NMY),JLR(NMY)
C
C    JNO(NMY): JUNCTION NUMBER.
C    T(NMY): TEMPERATURE OF AIR IN JUNCTIONS.
C    Z(NMY): ELEVATION OF JUNCTIONS.
C    CH4C(NMY): METHANE CONCENTRATION IN JUNCTION.
C    JNOL(NMY): LIST OF JUNCTION NUMBERS IN INCREASING ORDER.
C    PROP(NMY): CONTAMINANT CONCENTRATION IN JUNCTION.
C    PRCH4(NMY): METHANE CONCENTRATION IN JUNCTION.
C    JLR(NMY): LIST RELATING JNO- AND JNOL-LISTS.
C
C    COMMON  /TEMP/ TROCK(NMX),TMRD(NMX),HA(NMX),HK(NMX)
C
C    TROCK(NMX): AVERAGE ROCK TEMPERATURE IN AIRWAY.
C    TMRD(NMX): MEAN TEMPERATURE OF AIR IN AIRWAY.
C    HA(NMX): THERMAL DIFFUSIVITY OF ROCK.
C    HK(NMX): THERMAL CONDUCTIVITY OF ROCK.
C
C    COMMON  /FUME/ NCENT(NMX),CH4V(NMX),CH4PA(NMX),CONT(IMX),CONC(IMX)
C   .               ,HEAT(IMX),O2MIN(IMX),SMPO2(IMX),HTPO2(IMX),
C   .               TFSI(IMX),O2BEH(IMX),TPR(IMX),HTAD(IMX),QCENT(IMX)
C
C    NCENT(NMX): NUMBER OF AIRWAYS INTO WHICH FUME ENTERS.
C    CH4V(NMX): METHANE EMISSION RATE IN AIRWAYS.
C    CH4PA(NMX): METHANE EMISSION RATE PER UNIT SURFACE AREA.
C    CONT(IMX): VOLUME FLOW RATE OF CONTAMINATED GAS INFLOW.
C    CONC(IMX): CONCENTRATION OF CONTAMINANT IN GAS INFLOW.
C    HEAT(IMX): HEAT ENTERING AIRWAY.
C    O2MIN(IMX): OXYGEN CONCENTRATION OF FUMES LEAVING FIRE ZONE
C    SMPO2(IMX): CONTAMINANT PRODUCTION PER CUBIC FT OF OXYGEN.
C    HTPO2(IMX): HEAT PRODUCTION PER CUBIC FT OF OXYGEN DELIVERY
C    TFSI(IMX): AIR TEMPERATURE BEHIND HEAT SOURCE.
C    O2BEH(IMX): OXYGEN CONTENT BEHIND FIRE SOURCES.
C    TPR(IMX): TRANSITION TIME OF FIRE IN MIN.
C    HTAD(IMX): REDUCED HEAT INPUT DUE TO HIGH AIR TEMPERATURE.
C    QCENT(IMX): "STD. Q' AT WHICH FIRE TAKES THE INPUT PARAMETERS.
C
C    COMMON  /SCHEME/ NGIN(NMX),MMIN(NMX),NGOUT(NMX),LOUT(NMX)
C
C    NGIN(NMX): LIST OF AIRWAYS ENTERING JUNCTION.
C    MMIN(NMX): LIST OF LAST AIRWAY PER JUNCTION IN NGIN-LIST.
C    NGOUT(NMX): LIST OF AIRWAYS LEAVING JUNCTION.
C    LOUT(NMX): LIST OF LAST AIRWAY PER JUNCTION IN NGOUT-LIST.
C
C    COMMON  /EST/ MEMREC(NMX),NOREC(NMX),ESTPR(NMX),ESTCH4(NMX),
C   .              ESTTR(NMX)
C
C    MEMREC(NMX): TEMPORARY LIST OF AIRWAYS CARRYING RECIRC. AIR.
C    NOREC(NMX): PERMANENT LIST OF AIRWAYS.
C    ESTPR(NMX): ESTIMATED CONCENTRATION FOR RECIRCULATED AIR.
C    ESTCH4(NMX): ESTIMATED METHANE CONCENTRATION.
```

```
C    ESTTR(NMX): ESTIMATED TEMPERATURE.
C
C    COMMON  /FACTOR/ XNEW(NMX),DCOAGE(NMX)
C
C    XNEW(NMX): EXPONENT FOR TEMPERATURE CALCULATION.
C    DCOAGE(NMX): COEFFICIENT OF AGE.
C
C    COMMON  /TTJ/ TIME,TSTART,JSTART,TLEFT(NMX)
C
C    TIME: TIME ASSUMING QUASI-EQUILIBRIA.
C    TSTART: TEMPERATURE AT THE START JUNCTION.
C    JSTART: START JUNCTION (ATMOSPHERE).
C    TLEFT(NMX): TIME INCREMENT FOR MAKING UP A NORMAL INTERVAL.
C
C    COMMON  /SUM/ SUMQ(NMY),SUMC(NMY),SUMM(NMY),SMHEAT(NMY)
C
C    SUMQ(NMY): TOTAL AIRFLOW RATES ENTERING JUNCTION.
C    SUMC(NMY): TOTAL CONTAMINANT FLOW RATES ENTERING JUNCTION.
C    SUMM(NMY): TOTAL METHANE FLOW RATES ENTERING JUNCTION.
C    SMHEAT(NMY): TOTAL ENTHALPY/REFERENCE DENSITY ENTERING JUNCT.
C
C    COMMON  /AUX/ PROPS(NMX),PRCH4S(NMX),QQ(NMX),QTP(NMX),TTJS(NMX),
C   .              TTRD(NMX),TAS(NMX),BI(NMX)
C
C    PROPS(NMX): BACKUP ARRAY FOR PROP.
C    PRCH4S(NMX): BACKUP ARRAY FOR PRCH4.
C    QQ(NMX): DAMPED AND TIME-AVERAGED AIRFLOW RATE.
C    QTP(NMX): BACKUP ARRAY FOR Q, FIXED IN EACH INTERVAL.
C    TTJS(NMX): BACKUP ARRAY FOR TJS.
C    TTRD(NMX): BACKUP ARRAY FOR TRD.
C    TAS(NMX): BACKUP ARRAY OF T.
C    BI(NMX): BIOT NUMBER.
C
C    COMMON  /WRN/ WRNPR,WRNGS,WRNSM,WRNHT,IOMIT(IMX),ROMIT(IMX,3),IAC
C
C    WRNPR: PRESSURE DROP WARNING CRITERIA.
C    WRNGS: METHANE CONCENTRATION WARNING CRITERIA.
C    WRNSM: FUME CONCENTRATION WARNING CRITERIA.
C    WRNHT: HIGH TEMPERATURE WARNING CRITERIA.
C    IOMIT(IMX): AIRWAYS FOR WHICH DETAILED DATA RECORDS WERE OMITTED.
C    ROMIT(IMX,3): DATA AT STARTING J OF DATA-OMITTED AIRWAY.
C    IAC: COUNTER OF IOMIT AND ROMIT.
C
C    COMMON  /TRANS/ DELTAT,TACC,SUMFNV
C
C    DELTAT: SPAN OF TIME INCREMENT.
C    TACC: CUMULATED TIME AFTER EVENT IN SEC.
C    SUMFNV: SUM OF NVP CORRECTION IN SYSTEM.
C
C    COMMON  /RCD/ IBTN(NMX),DPPA(NMZ,10,5),NSAC(NMX,2),TMRDA(NMX),
C   .              DELTAQ(NMX),DPPB(NMZ,10,5),NSACB(NMX,2),GBTN(10,5),
C   .              JCH(NMY),TAUXA(NMZ,10),TAUXB(NMZ,10),TAUXC(NMZ),
C   .              TAUXD(NMZ),FMASS(NMZ)
C
```

```
C    IBTN(NMX): ARRAY HOLDING ADDRESS OF RECORD FOR AIRWAY (N
C    DPPA(NMZ,10,5): MASTER DATA STORAGE ARRAY IN DYNAMIC SIMULATION
C    NSAC(NMX,2): DATA RECORD STATUS OF AIRWAY (NMX).
C    TMRDA(NMX): BACKUP ARRAY FOR TMRD.
C    DELTAQ(NMX): VARIATION OF AIRFLOW IN AIRWAYS.
C    DPPB(NMZ,10,5): BACKUP ARRAY FOR DPPA.
C    NSACB(NMX,2): BACKUP ARRAY FOR NSAC.
C    GBTN(10,5): AUXILIARY ARRAY FOR DATA TRANSFER.
C    JCH(NMY): MARKER INDICATING JUNCTION CONDITION CHANGE.
C    TAUXA(NMZ,10): AIR TEMP. AT REAR ENDING OF SEGMENTS.
C    TAUXB(NMZ,10): BACKUP ARRAY FOR TAUXA.
C    TAUXC(NMZ): AIR TEMP. BEYOND THE FIRST SEGMENT.
C    TAUXD(NMZ): BACKUP ARRAY OF TAUXC.
C    FMASS(NMZ): MASS OF AIR BEYOND THE FIRST SEGMENT.
C
C    COMMON  /CCJ/ JDP, JDPP(IMX)
C
C    JDP: NUMBER OF JUNCTIONS INITIATING DATA RECORDS.
C    JDPP(IMX): JUNCTIONS WHICH INITIATE DATA RECORDS.
C
C    COMMON  /TREND/ QRCD(NMX,10),TMRCD(NMX,10),HRM(IMX)
C
C    QRCD(NMX,10): AIRFLOW RATES DURING ITERATIONS.
C    TMRCD(NMX,10): MEAN AIR TEMP. IN AIRWAYS DURING ITERATION.
C    HRM(IMX): VARIATION OF NVP PER MESH DURING ITERATION.
C
C    COMMON /CHRC/ ITITLE(50),NCOMT1,NCOMTS,NCOMT2,LLINE
C    COMMON /ERMSG/ ROW(80),TITLE(50,20)
C    COMMON /ATJUNT/JAN,JAJ,JA1(10),JATP(10),JSTAR(10),TSTAR(10),
C   .                    MSTAR(10),ISTAR(10)
C
C    JAN:  TOTAL NUMBER OF ATMOSPHERE JUNCTIONS
C    JAJ:  NUMBER OF ATMOSPHERE JUNCTIONS EXCEPT STARTING JUNCTION
C    JA1(10):  JUNCTION ID NUMBER (IN ATMOSPHERE)
C    JATP(10): JUNCTION TYPE (IN ATMOSPHERE)
C    JSTAR(10): JUNCTION CONNECTED WITH ENTRY AIRWAYS EXCEPT STARING
C                JUNCTION
C    TSTAR(10): TEMPERATURE IN THE JUNCTION AT ARRAY (JSTAR)
C    MSTAR(10): JUNCTION NUMBER CORRESPONDING TO ARRAY (JNOL)
C    ISTAR(10): JUNCTION NUMBER CORRESPONDING TO ARRAY (JNO)
C
C    REAL LA
C    CHARACTER*4 TITLE
C    CHARACTER*1 ROW
C
C
C    MARKX: 0: THE PROGRAM HAS NOT REACHED THE DYNAMIC SIMULATION
C             PART YET.   1: THE DYNAMIC PART HAS BEEN REACHED.
C    MARKY: 0: BEFORE FINAL QUASI-EQUILIBRIUM SIMULATION.
C             1: IN THE FINAL QUASI-EQUIL. STAGE, READY FOR TERMINATION.
C    NSTOP: MARKER FOR INPUT DATA ERROR.
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
     CHARACTER*50 MD1,MD2,OPTION
C-------------------------------------------------------------------------------------------------------------------------------
```

```
OPEN (10,FILE='NET',STATUS='UNKNOWN',FORM='UNFORMATTED')
READ (10) NETWW
IF (NETWW.EQ.1) GO TO 350
OPTION='                                                                    '
MD1='                                                                       '
MD2='                                                                       '
OPEN (21,FILE='ZHENG.CHG',STATUS='UNKNOWN',FORM='UNFORMATTED')
READ (21) OPTION
READ (21) MD1,TACC
IF (OPTION.EQ.'HUNAN-XIANGYING') GO TO 350
NSTOP=0
MARKX=0
IF (OPTION.EQ.'CHANGZHENGCHANGLIANG') THEN
   MARKY=1
   MD2='XINTAN.CHG'
   WRITE (6,740)
ELSE
   WRITE (6,910)
   PAUSE
   WRITE (6,900)
   PAUSE
   MARKY=0
   MD2='YAODONG.XIN'
   WRITE (6,730)
   WRITE (6,680)
   READ (5,'(A)') MD1
ENDIF
OPEN (9,FILE=MD1)
OPEN (8,FILE=MD2)
IF (MARKY.EQ.1) WRITE (8,760)
MADJC=0
LLINE=0
MAXNO=0
CALL INPUT (1,NSTOP,MARKY,MAXNO,KV)
IF (NSTOP.GT.0) GO TO 300
CALL CHECK1 (NSTOP,MAXNO,KV)
IF (NSTOP.GT.0) GO TO 300
IF (NETW.NE.1) THEN
   CALL INPUT (2,NSTOP,MARKY,MAXNO,KV)
   IF (NSTOP.GT.0) GO TO 300
ENDIF
ITCT=0
CALL ARR (MARKX)
CALL BASE
CALL MSLIST
CALL MBLNC
CALL NVP1
CALL ITR (MARKX,NSFLOW,0,MADJC,ITCT)
CALL RGLT (MARKY)
IF (MARKY.EQ.0) WRITE (6,670)
IF (NETW.EQ.1) THEN
   NETWW=1
   REWIND 10
   WRITE (10) NETWW
```

```
20
          WRITE (8,510)
          CALL OUTPUT (2,MARKX,MARKY,MADJC)
          GO TO 350
        ENDIF
C
C
        IF (MARKY.LE.0) THEN
          WRITE (6,855)
        ELSE
          WRITE (6,860)
        ENDIF
        WRITE (6,865)
        TAVR=0.0
        NSFLOW=0
        ITCT=0
        CALL INPUT (3,NSTOP,MARKY,MAXNO,KV)
        IF (NSTOP.GT.0) GO TO 300
        CALL CCDATA (MARKY,NSTOP)
        IF (NSTOP.GT.0) GO TO 300
        IF (MARKY.EQ.1) THEN
          CALL CDCH (NSTOP)
          IF (NSTOP.GT.0) GO TO 300
        ENDIF
        CALL CH4EVA
        IF ((IOUT.EQ.(-1).AND.MARKY.LE.0).OR.IOUT.LE.(-2))
     .    CALL OUTPUT (1,MARKX,MARKY,MADJC)
        IF ((IOUT.EQ.(-1).AND.MARKY.LE.0).OR.IOUT.LE.(-2))
     .    CALL OUTPUT (2,MARKX,MARKY,MADJC)
        CALL OUTPUT (3,MARKX,MARKY,MADJC)
        JAJ=0
        IF (JAN.LE.1) GO TO 40
          DO 10 J=1,JAN
            IF (JATP(J).NE.0) THEN
              JAJ=JAJ+1
              JSTAR(JAJ)=JA1(J)
            ENDIF
10        CONTINUE
        DO 30 J=1,NJ
          DO 30 K=1,JAJ
            IF (JSTAR(K).EQ.JNO(J)) THEN
              TSTAR(K)=T(J)
            ENDIF
30      CONTINUE
40      NSFLOW=1
        CALL FWCT (NSFLOW)
C
C       DETERMINATION OF CONDITIONS IN THE STARTING JUNCTION.
C
        DO 50 I=1,NJ
          PROP(I)=0.0
          PRCH4(I)=0.0
          IF (JNOL(I).EQ.JSTART) THEN
            MSTART=I
            ISTART=JLR(I)
```

```
            T(ISTART)=TSTART
          ENDIF
50      CONTINUE
70      MRC=0
        L=ISTART
        NSTART=MSTART
90      CALL CDENDS (MARKY,NSTART,L)
        JUMPBK=0
        CALL CDJUNC (NSTART,JUMPBK,L,0)
        IF (JUMPBK.EQ.1) GO TO 90
        CALL RECIRC (MRC,JUMPBK,NSTART,L)
        IF (JUMPBK.EQ.1) GO TO 90
        LST=0
        CALL PREP (MRC,LST)
        IF (MRC.NE.0) THEN
C
C       ITCT: COUNTER OF ITERATIONS.
C
          ITCT=ITCT+1
          IF (LST.NE.0) THEN
            IF (ITCT.LE.20) THEN
              L=ISTART
              NSTART=MSTART
              GO TO 90
            ENDIF
            WRITE (8,500) MADJC
          ENDIF
        ENDIF
        CALL TEVAL
C
C       NSFLOW: 0: DATA PREPARATION FOR NETWORK BALANCING IN TEMP. PART
C                  HAS NOT BEEN COMPLETED. AIRFLOW REVERSAL EXISTS IN THE
C                  SYSTEM.  1: DATA PREPARATION COMPLETED.
C       SUMFNV: SUM OF ABSOLUTE DIFFERENCE OF NVP IN SUCCESSIVE INTERVALS.
C       MADJ: USER-SPECIFIED MAX. NUMBER OF ITERATIONS IN ONE PROGRAM RUN.
C       MADJC: COUNTER OF ITERATIONS IN ONE PROGRAM RUN.
C
        IF (MADJC.LT.MADJ) THEN
          MADJC=MADJC+1
          ITCT=0
          DO 120 I=1,NB
            IF (JF(I).LT.0) JF(I)=-JF(I)
120       CONTINUE
          IF (NSFLOW.GT.0) THEN
            CALL NVP2
            CALL ITR (MARKX,NSFLOW,1,MADJC,ITCT)
          ELSE
            CALL ARR (MARKX)
            CALL BASE
            CALL MSLIST
            CALL MBLNC
            CALL NVP2
            CALL ITR (MARKX,NSFLOW,1,MADJC,ITCT)
          ENDIF
```

```
      WRITE (6,870) MADJC,SUMFNV/MNO
      IF ((SUMFNV/MNO).LE.2.E-4) GO TO 130
      IF (MADJC.GE.(MADJ-9)) THEN
        J=MADJC-MADJ+10
        HRM(J)=SUMFNV/MNO
        DO 125 I=1,NB
          QRCD(I,J)=ABS(Q(I))
          IF (JSB(I).NE.IABS(JS(I)).AND.Q(I).GT.0.0)
     .      QRCD(I,J)=-ABS(Q(I))
          IF (JSB(I).EQ.IABS(JS(I)).AND.Q(I).LT.0.0)
     .      QRCD(I,J)=-ABS(Q(I))
          TMRCD(I,J)=TMRD(I)
125     CONTINUE
      ENDIF
      ITCT=0
      DO 127 I=1,NB
        IF (Q(I).LT.0.0) GO TO 40
127   CONTINUE
      NSFLOW=1
      GO TO 70
    ENDIF
130 CALL FWCT (NSFLOW)
    IF (MARKY.LE.0) WRITE (8,620)
    CALL OUTPUT (4,MARKX,MARKY,MADJC)
    IF (IOUT.NE.0) CALL OUTPUT (5,MARKX,MARKY,MADJC)
    IF (NTEMP.EQ.1) WRITE (8,520)
    IF (MARKY.EQ.0) THEN
      WRITE (6,660)
    ELSE
      WRITE (6,750)
    ENDIF
C
C
    IF (NETW.GE.1.OR.NTEMP.GE.1) GO TO 300
    IF (OPTION.EQ.'CHANGZHENGCHANGLIANG') GO TO 350
    OPEN (20,FILE='LIANG.CHG',STATUS='UNKNOWN',FORM='UNFORMATTED')
    WRITE (20) NB,NJ,NFNUM,NETW,NTEMP,ITN,DR,TR,TINC,SPAN,IOUT,TOUT,
   .           CONCT,NSTOP
    WRITE (20) NAV,MAXJ,INFLOW,CRITSM,CRITGS,CRITHT,MNO
    WRITE (20) TIME,TSTART,JSTART,NSFLOW,WRNPR,WRNGS,WRNSM,WRNHT
    WRITE (20) MARKX,MARKY,NCOMT1,NCOMTS,NCOMT2,LLINE
    WRITE (20) JAN,JAJ
    DO 3200 I=1,NB
      WRITE (20) NO(I),JS(I),JF(I),NWTYP(I),R(I),Q(I),KF(I),LA(I)
      WRITE (20) A(I),O(I),RSTD(I),DZRD(I),FRNVP(I),NREV(I),RDCH4(I)
      WRITE (20) RDPROP(I),TRD(I),TJS(I),FFRNVP(I),NNREV(I),JSB(I)
      WRITE (20) TROCK(I),TMRD(I),HA(I),HK(I)
      WRITE (20) XNEW(I),DCOAGE(I),CH4V(I),CH4PA(I)
      WRITE (20) NGIN(I),MMIN(I),NGOUT(I),LOUT(I)
3200 CONTINUE
    DO 3202 I=1,IMX
      WRITE (20) NOF(I),NFREG(I),RGRAD(I),NFCW(I),MPTS(I),
     .           NSKP(I),NEGQ(I)
      WRITE (20) NCENT(I),CONT(I),CONC(I),HEAT(I),O2MIN(I),SMPO2(I),
```

```
     .                HTPO2(I),TPR(I),QCENT(I)
          DO 3201 J=1,IMY
              WRITE (20) QF(I,J),PF(I,J)
              DO 3205 K=1,4
                  WRITE (20) FKQ(I,J,K)
3205          CONTINUE
3201 CONTINUE
3202 CONTINUE
     DO 3203 I=1,NJ
          WRITE (20) JNO(I),T(I),Z(I),CH4C(I),JNOL(I),PROP(I),
     .                PRCH4(I),JLR(I)
3203 CONTINUE
     DO 3204 K=1,JAJ
          WRITE (20) JSTAR(K),TSTAR(K)
3204 CONTINUE
     DO 3206 I=1,MNO
          WRITE (20) MEND(I),FNVP(I)
3206 CONTINUE
     IYY=MEND(MNO)
     DO 3207 I=1,IYY
          WRITE (20) MSL(I)
3207 CONTINUE
     DO 3209 I=1,50
          WRITE (20) ITITLE(I)
          DO 3208 J=1,20
              WRITE (20) TITLE(I,J)
3208      CONTINUE
3209 CONTINUE
C
300  IF (MARKY.LE.0) THEN
     OPEN (21,FILE='ZHENG.CHG',STATUS='UNKNOWN',FORM='UNFORMATTED')
     OPTION='CHANGZHENGCHANGLIANG'
     IF (NSTOP.GE.1.OR.NETW.GE.1.OR.NTEMP.GE.1)
     .   OPTION='HUNAN-XIANGYING'
     REWIND (21)
     WRITE (21) OPTION
     WRITE (21) MD1,TACC
     IF (NSTOP.GE.1.AND.NSTOP.LE.4) THEN
        NNB=NB
        IF (NSTOP.EQ.1) NNB=KV
        WRITE (8,800)
        WRITE (8,810) (NO(I),JS(I),JF(I),NWTYP(I),R(I),Q(I),
     .                  KF(I),LA(I),A(I),O(I),I=1,NNB)
     ENDIF
     IF (NVPN.GE.1) WRITE (8,880)
     IF (NVPN.LE.0.AND.NSTOP.GE.2.AND.NSTOP.LE.4) THEN
        NNJ=NJ
        IF (NSTOP.EQ.2) NNJ=KV
        WRITE (8,820)
        WRITE (8,825) (JNO(J),T(J),Z(J),CH4C(J),J=1,NNJ)
     ENDIF
     IF (NFNUM.GT.0.AND.NSTOP.GE.3.AND.NSTOP.LE.4) THEN
        NNFNUM=NFNUM
        IF (NSTOP.EQ.3) NNFNUM=KV
```

```
      WRITE (8,830) (NOF(K),K=1,NNFNUM)
      WRITE (8,840)
      DO 310 K=1,NNFNUM
         L=MPTS(K)
         IF (L.GT.IMY) L=IMY
         DO 305 I=1,L
            QF(K,I)=QF(K,I)/1000.0
305      CONTINUE
         WRITE (8,850) (QF(K,I),PF(K,I),I=1,L)
310   CONTINUE
    ENDIF
    IF (NSTOP.GT.0) WRITE (8,890)
    ENDIF
350 STOP
C
500 FORMAT (//,' * ATTENTION *  DATA PREPARATION DID NOT COMPLETE ',
   .        'WITHIN',/,' 20 ITERATIONS IN DATA PREPARATION CYCLE',I4)
510 FORMAT (//,1X,'*****  NO FURTHER CALCULATIONS BEYOND NETWORK ',
   .        'BALANCING WERE',/,7X,'DEMANDED DUE TO NETW=1, PROG. ',
   .        'RUN COMPLETED.',/)
520 FORMAT (//,1X,'***** NO FURTHER CALCULATIONS BEYOND TEMP. ',
   .        'CALCULATIONS WERE',/,7X,'DEMANDED DUE TO NTEMP=1, PROG.',
   .        ' RUN COMPLETED.',/)
560 FORMAT (///,1X,71('*'),///)
600 FORMAT (8X,F8.4,T31,I5,T47,F6.2)
620 FORMAT (///,1X,71('*'),///,T13,'OUTPUT OF THE TEMPERATURE',
   .        ' PART OF THE PROGRAM',///,1X,71('*'))
660 FORMAT (/,1X,'TEMPERATURE PART COMPLETED.',/)
670 FORMAT (/,1X,'NETWORK CALCULATION PART COMPLETED.',/)
680 FORMAT (/,1X,'NAME YOUR INPUT FILE BY "D:\PATH\FNAME1.EXT".',//)
730 FORMAT (30(/),1X,'PROGRAM MFIRE ACTIVATED ...')
740 FORMAT (//,1X,'QUASI-EQUILIBRIUM PART STARTED.',/)
750 FORMAT (/,1X,'SIMULATION COMPLETED.',///)
760 FORMAT (///,1X,71('*'),///,T15,'OUTPUT OF THE QUASI-EQUILIBRIUM',
   .        ' SIMULATION PART',///,1X,71('*'))
800 FORMAT (////,1X,'THE FOLLOWING IS THE CRITICAL DATA READ IN ',
   .        'SO FAR:',///,1X,T4,'NO',T9,'JS',T14,'JF',T18,'NWTYP',
   .        T30,'R',T39,'Q',T46,'KF',T53,'LA',T61,'A',T69,'O',/)
810 FORMAT (4I5,T23,F10.3,F10.0,T44,I4,T49,F7.0,T58,F6.1,T66,F5.1)
820 FORMAT (///,T5,'JNO',T15,'T',T25,'Z',T35,'CH4C',/)
825 FORMAT (T4,I4,T11,F6.1,T21,F7.1,T31,F8.2)
830 FORMAT (//,' THESE CHARACTERISTICS WERE READ IN FOR FANS',5I5,/
   .        T45,5I5)
840 FORMAT (//,5(' Q*1000  PF '),/)
850 FORMAT (5(F8.1,F6.2)/,5(F8.1,F6.2),/)
855 FORMAT (/,1X,'ENTERING TEMPERATURE PART OF THE PROGRAM.')
860 FORMAT (/,1X,'ENTERING QUASI-EQUIL. PART OF THE PROGRAM.')
865 FORMAT (1X,'PREFIXED ACCURACY CRITERION = 0.0002 IN.W.G. PER ',
   .        'MESH.',/)
870 FORMAT (10X,'ITERATION ',I3,'  COMPLETED, CURRENT ERROR = ',
   .        F8.5)
880 FORMAT (/,1X,'NO JUNCTION CARDS ARE EXPECTED TO READ IN DUE TO ',
   .        'NVPN>1 OR =1.',/,1X,'ONLY THE NETWORK PART WILL BE ',
   .        'PERFORMED.')
```

```
890  FORMAT (//,1X,'PROGRAM RUN TERMINATED DUE TO INDICATED INPUT ',
     .        'ERROR.',/,' CORRECT THE ERROR THEN YOU MAY TRY AGAIN.',
     .        ///)
900  FORMAT (30(/),T20,16('-'),' MFIRE ',17('-'),//,T20,'A PROGRAM ',
     .        'FOR DYNAMIC SIMULATION OF MINE',/,T21,'VENTILATION ',
     .        'UNDER THERMAL DISTURBANCES',///,T24,'WRITTEN BY:',T43,
     .        'XINTAN CHANG',/T24,'SUPERVISED BY:',T39,'DR. R. E. ',
     .        'GREUER',//,T25,'OCT. 1988',T38,'AT MICHIGAN TECH',/,
     .        T20,40('-'),1(/),T26,'VERSION 1.20 JULY 1989')
910  FORMAT (30(/),T35,'DISCLAIMER',//,T10,'THE BUREAU OF MINES EXPRESS
     .LY DECLARES THAT THERE ARE NO',/,T10,'WARRANTIES EXPRESS OR IMPLIE
     .D WHICH APPLY TO THE SOFTWARE',/,T10,'CONTAINED HEREIN. BY ACCEPTA
     .NCE AND USE OF SAID SOFTWARE',/,T10,'WHICH IS CONVEYED TO THE USER
     .WITHOUT CONSIDERATION BY THE',/,T10,'BUREAU OF MINES, THE USER HE
     .ROF EXPRESSLY WAIVES ANY AND',/,T10,'ALL CLAIMS FOR DAMAGE AND/OR
     .SUITS FOR OR BY REASON OF',/,T10,'PERSONAL INJURY, OR PROPERTY DAM
     .AGE, INCLUDING SPECIAL',/,T10,'CONSEQUENTIAL OR OTHER SIMILAR DAMA
     .AGES ARISING OUT OF OR',/,T10,'IN ANY WAY CONNECTED WITH THE USE O
     .F THE SOFTWARE',/,T10,'CONTAINED HEREIN.',///)
C
      END
C
C
      BLOCK DATA
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C     SUBPROGRAM PURPOSE:
C     INITIATION OF THE VARIABLES AND ARRAYS IN COMMON BLOCKS.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C

      PARAMETER ( K1=NMX*4,   K2=NMX*3,   K3=NMX*6,      K4=NMX*8,    K5=NMY*3,
     .            K6=IMX*2,   K7=IMX*IMY*2,          K8=LMX+NMY+1, K9=NMY+NMX,
     .            K10=NMY*2,  K11=NMX*2+IMX*11,      K16=NMZ*10*5,  K22=NMX*2,
     .            K17=NMZ*10*5+NMX*2, K19=NMZ*22,    K21=IMX*3,     K23=NMY*4,
     .            K24=IMY*IMY+IMY*2+IMX*IMY*4 )
C
      DATA  NO,JS,JF,NWTYP,R,Q,P,KF,LA,A,O,RSTD,DZRD,FRNVP,NREV,RDCH4,
     .      RDPROP,TRD,TJS,RDOP,RCH4,FFRNVP,RA,NNREV
     .      /K1*0,K2*0.0,NMX*0,K3*0.0,NMX*0,K4*0.0,NMX*0/
      DATA  NOF,NFREG,RGRAD,NFCW,MPTS,QF,PF,NSKP,NEGQ,JSB
     .      /K6*0,IMX*0.0,K6*0,K7*0.0,IMX*0,IMX*0,NMX*0/
      DATA  FC,DK,FK,FKQ
     .      /K24*0.0/
      DATA  MNO,MEND,MSL,FNVP,RQ,INU,KNO,KJS,KJF
     .      /K8*0,K9*0.0,K1*0/
      DATA  JNO,T,Z,CH4C,JNOL,PROP,PRCH4,JLR
     .      /NMY*0,K5*0.0,NMY*0,K10*0.0,NMY*0/
      DATA  TROCK,TMRD,HA,HK
     .      /K1*0.0/
      DATA  NCENT,CH4V,CH4PA,CONT,CONC,HEAT,O2MIN,SMPO2,HTPO2,TFSI,
```

```
     .        O2BEH,TPR,HTAD,QCENT
     .        /NMX*0,K11*0.0/
      DATA  NGIN,MMIN,NGOUT,LOUT
     .        /K1*0/
      DATA  MEMREC,NOREC,ESTPR,ESTCH4,ESTTR
     .        /K22*0,K2*0.0/
      DATA  XNEW,DCOAGE
     .        /K22*0.0/
      DATA  TIME,TSTART,JSTART,TLEFT
     .        /0.0,0.0,0,NMX*0.0/
      DATA  SUMQ,SUMC,SUMM,SMHEAT
     .        /K23*0.0/
      DATA  PROPS,PRCH4S,QQ,QTP,TTJS,TTRD,TAS,BI
     .        /K4*0.0/
      DATA  DELTAT,TACC,SUMFNV
     .        /3*0.0/
      DATA  IBTN,DPPA,NSAC,TMRDA,DELTAQ,DPPB,NSACB,GBTN,JCH,TAUXA,
     .        TAUXB,TAUXC,TAUXD,FMASS
     .        /NMX*0,K16*0.0,K22*0,K17*0.0,K22*0,50*0.0,NMY*0,K19*0.0,
     .        NMZ*0.0/
      DATA  WRNPR,WRNGS,WRNSM,WRNHT,IOMIT,ROMIT,IAC
     .        /4*0.0,IMX*0,K21*0.0,0/
      DATA  ITITLE,NCOMT1,NCOMTS,NCOMT2,LLINE
     .        /54*0/
C
C
      END
C
C
      SUBROUTINE  INPUT (JUMP1,NSTOP,MARKY,MAXNO,K)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSE:
C        INPUT OF DATA.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
      DIMENSION DAL(15)
C
      IF (JUMP1.EQ.1) THEN
C
C
         NBR=0
         NCOMTS=0
         NCOMT1=0
10       CALL READIN (DAL,14,ISTOP,0)
         IF (ISTOP.EQ.1) THEN
            WRITE (8,520)
            WRITE (8,530) (ROW(IE),IE=1,80)
            NSTOP=5
            RETURN
```

```
      ENDIF
      IF (DAL(15).LT.(-1.E20)) GO TO 10
      IF (NCOMTS.GT.NCOMT1) THEN
         NCOMT1=NCOMTS
         GO TO 10
      ENDIF
      NB=DAL(1)
      NFNUM=DAL(2)
      INFLOW=DAL(3)
      NVPN=DAL(4)
      NETW=DAL(5)
      TR=DAL(6)
      MADJ=DAL(7)
      ITN=DAL(8)
      NTEMP=DAL(9)
      TINC=DAL(10)
      SPAN=DAL(11)
      IOUT=DAL(12)
      TOUT=DAL(13)
      CONCT=DAL(14)
      IF (MARKY.EQ.0) WRITE (6,720)
      IF (ABS(TR-60.).GT.50.) TR=75.0
      DR=2112.6/(53.35*(TR+460.0))
      IF (MADJ.LT.5.OR.MADJ.GT.80) MADJ=10
      IF (ITN.LT.5.OR.ITN.GT.80) ITN=10
      IF (NB.GT.NMX) WRITE (8,210) NB,NMX
      IF (NB.GT.NMX) THEN
         WRITE (8,520)
         WRITE (8,530) (ROW(IE),IE=1,80)
         NSTOP=5
         RETURN
      ENDIF
      IF (IOUT.GT.0) IOUT=1
      IF (NVPN.GE.1) NETW=1
      III=1
      KNT=NB
      IF (NB.LE.1) KNT=NMX
      NCOMT2=NCOMTS
      DO 30 K=1,KNT
20       CALL READIN (DAL,15,ISTOP,1)
         IF (ISTOP.EQ.1) THEN
            WRITE (8,540)
            WRITE (8,550) K,(ROW(IE),IE=1,80)
            NSTOP=1
            RETURN
         ENDIF
         IF (DAL(15).LT.(-1.E20)) GO TO 20
         IF (NCOMTS.GT.NCOMT2) THEN
            NCOMT2=NCOMTS
            IF (NCOMTS.LE.50) ITITLE(NCOMTS)=K
            GO TO 20
         ENDIF
         IF (DAL(1).LT.0.0.OR.DAL(2).LT.0.0.OR.DAL(3).LT.0.0.OR.
     .       DAL(7).LT.0.0.OR.DAL(8).LT.0.0.OR.DAL(9).LT.0.0.OR.
```

```
          DAL(10).LT.0.0) THEN
         WRITE (8,380) INT(DAL(1))
         WRITE (8,540)
         WRITE (8,550) K,(ROW(IE),IE=1,80)
         NSTOP=1
         RETURN
ENDIF
IF (DAL(2).GE.1000.0.OR.DAL(3).GE.1000.0) THEN
   WRITE (8,390) INT(DAL(1))
   WRITE (8,540)
   WRITE (8,550) K,(ROW(IE),IE=1,80)
   NSTOP=1
   RETURN
ENDIF
IF (ABS(DAL(4)).GT.1.1) THEN
   WRITE (8,400) INT(DAL(1))
   WRITE (8,540)
   WRITE (8,550) K,(ROW(IE),IE=1,80)
   NSTOP=1
   RETURN
ENDIF
NO(K)=DAL(1)
JS(K)=DAL(2)
JF(K)=DAL(3)
NWTYP(K)=DAL(4)
R(K)=DAL(5)
Q(K)=DAL(6)
KF(K)=DAL(7)
LA(K)=DAL(8)
A(K)=DAL(9)
O(K)=DAL(10)
IF (DAL(11).GT.1.E-3) HA(K)=DAL(11)
IF (DAL(12).GT.1.E-3) HK(K)=DAL(12)
IF (DAL(13).GT.1.E-3) CH4V(K)=DAL(13)
IF (DAL(14).GT.1.E-3) CH4PA(K)=DAL(14)
IF (DAL(15).GT.1.E-3) TROCK(K)=DAL(15)
IF (NWTYP(K).EQ.(-1).AND.Q(K).LT.1.0) THEN
   WRITE (8,510) NO(K),Q(K)
   WRITE (8,540)
   WRITE (8,550) K,(ROW(IE),IE=1,80)
   NSTOP=1
   RETURN
ENDIF
IF (NO(K).LT.9990) THEN
   NBR=NBR+1
   K1=JS(K)
   MSL(K1)=1
   K1=JF(K)
   MSL(K1)=1
ELSE
   GO TO 40
ENDIF
NODE=0
DO 30 I=1,999
```

```
            IF (MSL(I).EQ.1) NODE=NODE+1
            IF (K.EQ.NB.OR.NO(K).GT.9990) MSL(I)=0
30        CONTINUE
          IF (NB.LE.1.AND.NO(K).LE.9990) THEN
            K=NMX
            WRITE (8,260) NMX,NO(K)
            NSTOP=1
            RETURN
          ENDIF
40        NCOMTS=100
          NCOMT2=NCOMTS
          IF (NBR.LE.1) THEN
            WRITE (8,360)
            NSTOP=1
            RETURN
          ENDIF
          IF (NB.LE.1.OR.NB.GT.NBR) NB=NBR
          IF (MARKY.EQ.0) WRITE (6,730) NB
          DO 50 I=1,NB
            JSB(I)=JS(I)
50        CONTINUE
C
C     NVPN: MARKER FOR PRESENCE OF JUNCTION CARDS.
C
          IF (III.GT.50.AND.MARKY.LE.0) WRITE (8,255) III
          NJ=NODE
          IF (NJ.GT.NMY) THEN
            WRITE (8,220) NJ,NMY
            NSTOP=5
            RETURN
          ENDIF
          IF (NVPN.LE.0) THEN
            DO 70 K=1,NJ
60            CALL READIN (DAL,4,ISTOP,0)
              IF (ISTOP.EQ.1) THEN
                WRITE (8,560)
                WRITE (8,570) K,(ROW(IE),IE=1,80)
                NSTOP=2
                RETURN
              ENDIF
              IF (DAL(15).LT.(-1.E20)) GO TO 60
              IF (NCOMTS.GT.NCOMT2) THEN
                NCOMT2=NCOMTS
                GO TO 60
              ENDIF
              IF (DAL(1).LE.0.0.OR.DAL(1).GE.1000.0) THEN
                WRITE (8,410) INT(DAL(1))
                WRITE (8,560)
                WRITE (8,570) K,(ROW(IE),IE=1,80)
                NSTOP=2
                RETURN
              ENDIF
              IF (DAL(4).LT.0.0.OR.DAL(4).GE.100.0) THEN
                WRITE (8,830) INT(DAL(1))
```

```
                WRITE (8,560)
                WRITE (8,570) K,(ROW(IE),IE=1,80)
                NSTOP=2
                RETURN
              ELSE IF (DAL(4).GT.5.0.AND.MARKY.LE.0) THEN
                WRITE (8,860) DAL(4),INT(DAL(1))
              ENDIF
              JNO(K)=DAL(1)
              IF (ABS(DAL(2)-75.).GT.50.0.AND.MARKY.LE.0)
                WRITE (8,420) DAL(2),JNO(K)
              T(K)=DAL(2)
              Z(K)=DAL(3)
              CH4C(K)=DAL(4)
70          CONTINUE
            IF (MARKY.EQ.0) WRITE (6,740) NJ
          ENDIF
72        CALL READIN (DAL,1,ISTOP,0)
          IF (ISTOP.EQ.1) THEN
            WRITE (8,575)
            WRITE (8,576) (ROW(IE),IE=1,80)
            NSTOP=1
            RETURN
          ENDIF
          IF (DAL(15).LT.(-1.E20)) GO TO 72
          IF (NCOMTS.GT.NCOMT2) THEN
            NCOMT2=NCOMTS
            GO TO 72
          ENDIF
          JAN=DAL(1)
          DO 75 I=1,JAN
73          CALL READIN (DAL,2,ISTOP,0)
            IF (ISTOP.EQ.1) THEN
              WRITE (8,577)
              WRITE (8,578) (ROW(IE),IE=1,80)
              NSTOP=1
              RETURN
            ENDIF
          IF (DAL(15).LT.(-1.E20)) GO TO 73
          IF (NCOMTS.GT.NCOMT2) THEN
            NCOMT2=NCOMTS
            GO TO 73
          ENDIF
            JA1(I)=DAL(1)
            JATP(I)=DAL(2)
75        CONTINUE
C
C     NFNUM: NUMBER OF FANS WITH THEIR CHARACTERISTICS.
C
          IF (NFNUM.GT.0) THEN
            IF (NFNUM.GT.IMX) THEN
              WRITE (8,230) NFNUM,IMX
              NSTOP=5
            ENDIF
            DO 110 K=1,NFNUM
```

```
80          CALL READIN (DAL,2,ISTOP,0)
            IF (ISTOP.EQ.1) THEN
               WRITE (8,580)
               WRITE (8,590) K,(ROW(IE),IE=1,80)
               NSTOP=3
               RETURN
            ENDIF
            IF (DAL(15).LT.(-1.E20)) GO TO 80
            IF (NCOMTS.GT.NCOMT2) THEN
               NCOMT2=NCOMTS
               GO TO 80
            ENDIF
            NOF(K)=DAL(1)
            MPTS(K)=DAL(2)
            INDEX=MPTS(K)
            IF (INDEX.LE.1.OR.INDEX.GT.IMY) THEN
               WRITE (8,240) INDEX,IMY
               WRITE (8,580)
               WRITE (8,590) K,(ROW(IE),IE=1,80)
               NSTOP=3
               RETURN
            ENDIF
            DO 100 JDEX=1,INDEX,5
               I=10
               IF ((JDEX+4).GT.INDEX) I=((INDEX+1)-JDEX)*2
90             CALL READIN(DAL,I,ISTOP,0)
               IF (ISTOP.EQ.1) THEN
                  WRITE (8,600)
                  WRITE (8,610) K,(ROW(IE),IE=1,80)
                  NSTOP=3
                  RETURN
               ENDIF
               IF (DAL(15).LT.(-1.E20)) GO TO 90
               IF (NCOMTS.GT.NCOMT2) THEN
                  NCOMT2=NCOMTS
                  GO TO 90
               ENDIF
               IF (INDEX.GE.(JDEX+4)) THEN
                  PF(K,JDEX+4)=DAL(10)
                  QF(K,JDEX+4)=DAL(9)
               ENDIF
               IF (INDEX.GE.(JDEX+3)) THEN
                  PF(K,JDEX+3)=DAL(8)
                  QF(K,JDEX+3)=DAL(7)
               ENDIF
               IF (INDEX.GE.(JDEX+2)) THEN
                  PF(K,JDEX+2)=DAL(6)
                  QF(K,JDEX+2)=DAL(5)
               ENDIF
               IF (INDEX.GE.(JDEX+1)) THEN
                  PF(K,JDEX+1)=DAL(4)
                  QF(K,JDEX+1)=DAL(3)
               ENDIF
               PF(K,JDEX)=DAL(2)
```

```
                QF(K,JDEX)=DAL(1)
100           CONTINUE
              LSTOP=0
              DO 105 J=1,INDEX-1
                IF (QF(K,J).GE.QF(K,J+1)) LSTOP=1
                IF (QF(K,J).LT.0.0) THEN
                  WRITE (8,920) NOF(K)
                  NSTOP=3
                  RETURN
                ENDIF
105           CONTINUE
              IF (LSTOP.EQ.1) THEN
                WRITE (8,430) NOF(K)
                NSTOP=3
                RETURN
              ENDIF
110         CONTINUE
          ENDIF
          DO 120 I=1,NB
            IF (NWTYP(I).EQ.1) THEN
              IF (NFNUM.GT.0) THEN
                DO 115 J=1,NFNUM
                  IF (NO(I).EQ.NOF(J)) GO TO 120
115             CONTINUE
              ENDIF
              IF (R(I).LT.0.0) THEN
                WRITE (8,810) NO(I),R(I)
                NSTOP=5
                RETURN
              ELSE IF (R(I).GT.18.0.AND.MARKY.LE.0) THEN
                WRITE (8,820) NO(I),R(I)
              ENDIF
            ENDIF
120       CONTINUE
          IF (MARKY.EQ.0) WRITE (6,750) NFNUM
          NCOMTT=NCOMT1
          IF (NCOMT1.GT.50) NCOMTT=50
          IF (NCOMT1.GT.0.AND.NSTOP.EQ.0.AND.NETW.GE.1.
     .      AND.MARKY.LE.0)THEN
            WRITE (8,201)
            WRITE (8,203) ((TITLE(I,J),J=1,18),I=1,NCOMTT)
            WRITE (8,201)
            IF (NVPN.GE.1) WRITE (8,270)
          ENDIF
        ELSE IF (JUMP1.EQ.2) THEN
C
C       INFLOW: NUMBER OF CONTAMINATION CARDS.
C
          IF (INFLOW.GT.0) THEN
            IF (INFLOW.GT.IMX) THEN
              WRITE (8,250) INFLOW,IMX
              NSTOP=5
              RETURN
            ENDIF
```

```
      IF ((NB+INFLOW).GT.NMX) THEN
        WRITE (8,340) NB,INFLOW,NMX
        NSTOP=5
        RETURN
      ELSE
        DO 160 I=1,INFLOW
140       CALL READIN (DAL,9,ISTOP,0)
          IF (ISTOP.EQ.1) THEN
            K=I
            WRITE (8,660)
            WRITE (8,670) I,(ROW(IE),IE=1,80)
            NSTOP=4
            RETURN
          ENDIF
          IF (DAL(15).LT.(-1.E20)) GO TO 140
          IF (NCOMTS.GT.NCOMT2) THEN
            NCOMT2=NCOMTS
            GO TO 140
          ENDIF
          DO 142 J=1,NB
            IF (INT(DAL(1)).EQ.NO(J)) GO TO 143
142       CONTINUE
          IF (MARKY.LE.0) WRITE (8,470) INT(DAL(1))
          GO TO 160
143       IF (DAL(2).LT.0.0.OR.DAL(3).LT.0.0.OR.DAL(5).LT.0.0.
     .      OR.DAL(6).LT.0.0.OR.DAL(7).LT.0.0) THEN
            K=I
            WRITE (8,840)
            WRITE (8,660)
            WRITE (8,670) I,(ROW(IE),IE=1,80)
            NSTOP=4
            RETURN
          ENDIF
          IF (DAL(3).GT.100.0) THEN
            K=I
            WRITE (8,880) DAL(3),INT(DAL(1))
            WRITE (8,660)
            WRITE (8,670) I,(ROW(IE),IE=1,80)
            NSTOP=4
            RETURN
          ENDIF
          NCENT(I)=DAL(1)
          CONT(I)=DAL(2)
          CONC(I)=DAL(3)
          HEAT(I)=DAL(4)
          O2MIN(I)=DAL(5)
          SMPO2(I)=DAL(6)
          HTPO2(I)=DAL(7)
          QCENT(I)=DAL(8)
          TPR(I)=DAL(9)
          IF (TPR(I).LT.1.E-5) TPR(I)=1.E-5
          IF (QCENT(I).LT.0.0) QCENT(I)=0.0
          IF (QCENT(I).GT.10.0) THEN
            VART=(9900.0+TR)**2+2.0*HEAT(I)/
```

```
     .                  (QCENT(I)*DR*2.4E-5)
                    TESTI=-9900.0+SQRT(VART)
                    IF (O2MIN(I).LE.0.1.AND.SMPO2(I).LE.1.E-3.AND.
     .              HTPO2(I).LT.1.0.AND.TESTI.GT.3000.0.AND.MARKY.LE.0)
     .                  WRITE (8,870) NCENT(I)
                  ENDIF
                  DO 145 K=1,NJ
                      IF (JS(J).EQ.JNO(K)) GO TO 150
145               CONTINUE
                  WRITE (8,900) NCENT
                  NCENT(I)=0
                  GO TO 160
150               DO 155 I2=NB+1,J+1,-1
                      NO(I2)=NO(I2-1)
                      JS(I2)=JS(I2-1)
                      JF(I2)=JF(I2-1)
                      NWTYP(I2)=NWTYP(I2-1)
                      R(I2)=R(I2-1)
                      RSTD(I2)=RSTD(I2-1)
                      Q(I2)=Q(I2-1)
                      KF(I2)=KF(I2-1)
                      LA(I2)=LA(I2-1)
                      A(I2)=A(I2-1)
                      O(I2)=O(I2-1)
                      HA(I2)=HA(I2-1)
                      HK(I2)=HK(I2-1)
                      CH4V(I2)=CH4V(I2-1)
                      CH4PA(I2)=CH4PA(I2-1)
                      TROCK(I2)=TROCK(I2-1)
155               CONTINUE
                  NO(J+1)=1000+MAXNO+1
                  IF (NO(J+1).GT.9995) NO(J+1)=MAXNO+1
                  JS(J+1)=MAXJ+1
                  JF(J)=MAXJ+1
                  NWTYP(J)=10
                  RSTD(J+1)=R(J+1)
                  R(J)=R(J)*0.1/LA(J)
                  IF (R(J).GT.0.001) R(J)=0.001
                  RSTD(J)=R(J)
                  LA(J+1)=LA(J)
                  LA(J)=0.1
                  CH4V(J)=0.0
                  CH4PA(J)=0.0
                  JNO(NJ+1)=MAXJ+1
                  T(NJ+1)=T(K)
                  Z(NJ+1)=Z(K)
                  CH4C(NJ+1)=CH4C(K)
                  DO 159 I3=1,50
                      IF (ITITLE(I3).GT.J) ITITLE(J)=ITITLE(J)+1
159               CONTINUE
                  NB=NB+1
                  NJ=NJ+1
                  MAXNO=MAXNO+1
                  MAXJ=MAXJ+1
```

```
160            CONTINUE
               IF (MARKY.EQ.0) WRITE (6,760) INFLOW
               DO 165 I=1,NB
                  JSB(I)=JS(I)
165            CONTINUE
             ENDIF
           ENDIF
           NCOMTT=NCOMT1
           IF (NCOMT1.GT.50) NCOMTT=50
           IF (NSTOP.EQ.0) THEN
             IF (MARKY.LE.0) WRITE (8,910)
             IF (NCOMT1.GT.0.AND.MARKY.LE.0) THEN
               WRITE (8,201)
               WRITE (8,203) ((TITLE(I,J),J=1,18),I=1,NCOMTT)
               WRITE (8,201)
             ENDIF
             IF (MARKY.LE.0.OR.IOUT.LE.(-2)) WRITE (8,280) TR,DR
             IF (INFLOW.GT.0.AND.MARKY.EQ.0) THEN
               WRITE (8,350)
               DO 167 I=1,INFLOW
                  IF (HTPO2(I).GE.500.) WRITE (8,890) HTPO2(I),NCENT(I)
167            CONTINUE
             ENDIF
           ENDIF
C
C
      ELSE
170        CALL READIN (DAL,11,ISTOP,0)
           IF (ISTOP.EQ.1) THEN
             WRITE (8,640)
             WRITE (8,650) (ROW(IE),IE=1,80)
             NSTOP=5
             RETURN
           ENDIF
           IF (DAL(15).LT.(-1.E20)) GO TO 170
           IF (NCOMTS.GT.NCOMT2) THEN
             NCOMT2=NCOMTS
             GO TO 170
           ENDIF
           NAV=DAL(1)
           JSTART=DAL(2)
           TSTART=DAL(3)
           TIME=DAL(4)
           CRITSM=DAL(5)
           CRITGS=DAL(6)
           CRITHT=DAL(7)
           WRNPR=DAL(8)
           WRNSM=DAL(9)
           WRNGS=DAL(10)
           WRNHT=DAL(11)
           IF (TIME.LT.1.E-3) TIME=5.0
           IF (CRITSM.LT.1.E-4.OR.CRITSM.GT.1.0) CRITSM=0.005
           IF (CRITGS.LT.1.E-4.OR.CRITGS.GT.1.0) CRITGS=0.01
           IF (CRITHT.LT.1.E-4.OR.CRITHT.GT.1.0) CRITHT=0.1
```

```
      IF (WRNPR.LE.1.E-10.OR.WRNPR.GT.0.5) WRNPR=0.01
      IF (WRNSM.LE.1.E-10.OR.WRNSM.GT.1.0) WRNSM=0.05
      IF (WRNGS.LE.1.E-10.OR.WRNGS.GT.10.0) WRNGS=1.0
      IF (WRNHT.LE.1.E-10.OR.WRNHT.GT.1000.0) WRNHT=100.0
      IF (MARKY.EQ.0) WRITE (6,780)
      J3=0
      DO 175 I=1,NJ
         IF (JNO(I).EQ.JSTART) J3=1
175   CONTINUE
      IF (J3.EQ.0) THEN
         WRITE (8,330) JSTART
         WRITE (8,640)
         WRITE (8,650) (ROW(IE),IE=1,80)
         NSTOP=5
         RETURN
      ENDIF
      IF ((TSTART.LT.-40.0.OR.TSTART.GT.120.0).AND.MARKY.LE.0)
     .   WRITE (8,850) TSTART
      IF (TIME.LE.0.0) THEN
         WRITE (8,460) TIME
         WRITE (8,640)
         WRITE (8,650) (ROW(IE),IE=1,80)
         NSTOP=5
         RETURN
      ENDIF
   ENDIF
   RETURN
C
200  FORMAT (20A4)
201  FORMAT (///)
203  FORMAT (18A4)
210  FORMAT (/,1X,'* ERROR *  NETWORK CONTAINS',I4,' AIRWAYS',/,
     .        1X,'WHICH IS BEYOND THE CAPACITY OF ARRAYS:',I5)
220  FORMAT (/,1X,'* ERROR *  NETWORK CONTAINS',I4,' JUNCTIONS',/,
     .        1X,'WHICH IS BEYOND THE CAPACITY OF ARRAYS:',I5)
230  FORMAT (/,1X,'* ERROR *  NETWORK CONTAINS',I4,' FANS',/,
     .        1X,'WHICH IS BEYOND THE CAPACITY OF ARRAYS:',I4)
240  FORMAT (/,1X,'* ERROR *  FAN CURVE CONTAINS',I4,' POINTS',/,
     .        1X,'WHICH IS BEYOND THE CAPACITY OF ARRAYS:',I4)
250  FORMAT (/,1X,'* ERROR *  NETWORK CONTAINS',I4,' FIRE SOURCES'
     .        ,/,1X,'WHICH IS BEYOND THE CAPACITY OF ARRAYS:',I4)
255  FORMAT (/,1X,'* ATTENTION *  NUMBER OF TITLES:',I3,' EXCEEDS THE '
     .        ,'MAXIMUM (50).',/,' TITLE AFTER NO. 50 WILL BE IGNORED.')
260  FORMAT (/,1X,'* ERROR *  MAXIMUM ALLOWED BRANCHES (',I3,') WAS ',
     .        'READ IN WITHOUT FINDING',/,' ENDING AIRWAY (9999). LAST'
     .        ,' BRANCH READ IN WAS (',I5,')')
270  FORMAT (/,1X,'NO JUNCTION CARDS ARE EXPECTED TO READ IN DUE TO ',
     .        'NVPN>1 OR =1.',/,1X,'ONLY THE NETWORK PART WILL BE ',
     .        'PERFORMED.')
280  FORMAT (/////,' REF.TEMP.(TR): ',F6.1,' DEG.F',11X,'REF. DENSITY',
     .        ' (DR): ',F6.4,' LBM/FT3',///,' LIST OF UNITS USED IN ',
     .        'THIS OUTPUT FILE:',//,' LENGTH:  FT.;',T25,'AREA:  FT2;'
     .        ,T43,'VOLUME:  FT3;',T61,'MASS:  LBM;',/,' AIRFLOW: ',
     .        'FT3/MIN;',T25,'TEMP.  F;',T43,
```

```
     .          'RESISTANCE:',6X,'IN.W.G./CFM2;',/,' FRICTION FACTOR:',7X,
     .          'LBM*MIN2/FT4;',T43,'THERMAL DIFFUSIVITY:   FT2/HR;',/,1X,
     .          'THERMAL CONDUCTIVITY:  BTU/HR*FT*F;',T43,'CONCENTRATION:'
     .          ,5X,'PERCENTAGE;',/,' HEADLOSS:',14X,'IN.W.G.;',T43,
     .          'HEAT INPUT:',11X,'BTU/MIN;',/,' CH4 PRODUCTION:',8X,
     .          'FT3/MIN;',///)
  330 FORMAT (/,1X,'START JUNCTION (JSTART) ',I3,' NOT FOUND.')
  340 FORMAT (/,1X,'* ERROR *  CURRENT INPUT DATA:  NB ',I4,
     .          ' AND INFLOW ',I3,/,1X,'THE SUM OF NB AND INFLOW ',
     .          'MUST BE LESS THAN',I6)
  350 FORMAT (/,1X,'* ATTENTION *  A NEW BRANCH HAS BEEN SET UP FOR ',
     .          'EACH FIRE SOURCE.')
  360 FORMAT (/,1X,'* ERROR *  TOO FEW AIRWAY CARD INPUT. AIRWAY ID ',
     .          ' NUMBER > 9990',/,' MARKS THE ENDING OF AIRWAY DATA.')
  380 FORMAT (/,1X,'* ERROR *  ILLEGAL NEGATIVE INPUT DETECTED FOR ',
     .          'AIRWAY ',I4)
  390 FORMAT (/,1X,'* ERROR *  JUNCTION ID > = 1000 DETECTED FOR ',
     .          'AIRWAY ',I4)
  400 FORMAT (/,1X,'* ERROR *  ILLEGAL INPUT VALUE FOR NWTYP DETECTED',
     .          ' FOR AIRWAY ',I4)
  410 FORMAT (/,1X,'* ERROR *  JUNCTION ID > = 1000 OR <0 DETECTED IN ',
     .          'JUNCTION',/,1X,'CARDS AS ',I5)
  420 FORMAT (/,1X,'* ATTENTION *  CHECK UNLIKELY VALUE OF TEMPERATURE',
     .          F7.1,/,' FOR JUNCTION ',I4)
  430 FORMAT (/,1X,'* ERROR *  FAN CURVE INPUT MUST BE IN THE ORDER ',
     .          'FROM LOW-QF',/,' TO HIGH-QF. INVALID ORDER DETECTED ',
     .          'FOR FAN ',I4)
  460 FORMAT (/,1X,'* ERROR *  ILLEGAL ZERO OR NEGATIVE INPUT VALUE ',
     .          /,' OF TIME ',F7.1,' (HOURS) DETECTED.')
  470 FORMAT (/,1X,'* ATTENTION *  FIRE SOURCE ',I5,' DOES NOT MATCH ',
     .          'ANY AIRWAY',/,' IN THE NETWORK, DATA IGNORED.')
  510 FORMAT (//,' * ERROR *  THE SPECIFIED FIXED AIRFLOW FOR ',
     .          'AIRWAY ',I4,' WAS',/,' UNLIKELY SMALL OR NEGATIVE. ',
     .          ' INPUT OF Q WAS ',F12.1)
  520 FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .          'IN:',//,1X,'NB,NFNUM,INFLOW,NVPN,NETW,TR,MADJ,ITN')
  530 FORMAT (/,1X,'THE CURRENT INPUT WAS:',//,80A1)
  540 FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .          'IN:',//,1X,'NO,JS,JF,NWTYP,R,Q,KF,LA,A,O,TROCK,HA,HK,',
     .          'CH4V,CH4PA')
  550 FORMAT (/,1X,'THE CURRENT INPUT FOR AIRWAY ',I4,' WAS:',//,80A1)
  560 FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .          'IN:',//,1X,'JNO,T,Z,CH4C')
  570 FORMAT (/,1X,'THE CURRENT INPUT FOR JUNCTION ',I4,' WAS:',//,80A1)
  575 FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .          'IN:',//,1X,'JAN')
  576 FORMAT (/,1X,'THE CURRENT INPUT FOR JUNCTION ',I4,' WAS:',//,80A1)
  577 FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .          'IN:',//,1X,'JA,JATP')
  578 FORMAT (/,1X,'THE CURRENT INPUT FOR JUNCTION ',I4,' WAS:',//,80A1)
  580 FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .          'IN:',//,1X,'NOF,MPTS')
  590 FORMAT (/,' THE CURRENT INPUT FOR FAN ',I4,' WAS:',//,80A1)
  600 FORMAT (/,' THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
```

```
     .          'IN:',//,' QF,PF, QF,PF, QF,PF, QF,PF, QF,PF')
610  FORMAT (/,' THE CURRENT INPUT FOR FAN ',I4,' WAS:',//,80A1)
640  FORMAT (/,' THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .          'IN:',//,' NAV,JSTART,TSTART,TIME,CRITSM,CRITGS,CRITHT,',
     .          'WRNPR,WRNSM,WRNGS,WRNHT')
650  FORMAT (/,' THE CURRENT INPUT FOR THE SECOND CONTROL CARD WAS:',
     .          //,80A1)
660  FORMAT (/,' THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .          'IN:',//,' NCENT,CONT,CONC,HEAT,O2MIN,SMPO2,HTPO2,QCENT')
670  FORMAT (/,' THE CURRENT INPUT FOR FIRE SOURCE ',I4,' WAS:',//,
     .          80A1)
720  FORMAT (/,5X,'CONTROL CARD 1 WAS READ IN.')
730  FORMAT (5X,'TOTAL',I4,' AIRWAY CARDS WERE READ IN.')
740  FORMAT (5X,'TOTAL',I4,' JUNCTION CARDS WERE READ IN.')
750  FORMAT (5X,'TOTAL',I3,' FAN CURVE(S) WERE READ IN')
760  FORMAT (5X,'TOTAL',I3,' FIRE SOURCE CARD(S) WERE READ IN.')
780  FORMAT (5X,'CONTROL CARD 2 WAS READ IN.',/)
810  FORMAT (/,1X,'* ERROR *  FIXED-PRESSURE FAN',I5,' HAS AN ILLEGAL'
     .          ,/,1X,'NEGATIVE PRESSURE VALUE ',F7.3)
820  FORMAT (/,1X,'* WARNING *  THE INPUT VALUE FOR FIXED-PRESSURE ',
     .          'FAN',I5,' WAS',/,1X,'UNLIKELY TOO LARGE. THE INPUT ',
     .          'FAN PRESSURE WAS ',F9.3)
830  FORMAT (/,1X,'* ERROR *  JUNCTION ',I4,' HAS AN INVALID CH4 ',
     .          'INPUT. IT MUST BE',/,' WITHIN 0.0 (NO CH4) AND 100.0',
     .          ' (PURE CH4).')
840  FORMAT (/,1X,'* ERROR *  ILLEGAL NEGATIVE INPUT DETECTED IN ',
     .          'READING FIRE SOURCE CARD.')
850  FORMAT (/,1X,'* ATTENTION *  CHECK UNLIKELY VALUE OF AIR TEMP. ',
     .          'AT STARTING JUNCTION.',/,' THE CURRENT INPUT VALUE WAS',
     .          F9.2,' (F).')
860  FORMAT (/,1X,'* WARNING *  AN UNLIKELY INPUT OF CH4 CONCENTRATION'
     .          ,F6.1,' %',/,' AT JUNCTION ',I3,' DETECTED.')
870  FORMAT (/,1X,'* WARNING *  FIXED-HEAT-INPUT MODEL WAS SPECIFIED ',
     .          'FOR FIRE IN',/,' AIRWAY',I5,'. BUT QCENT WAS TOO SMALL ',
     .          'OR "HEAT" WAS TOO LARGE.',/,' EXCEPTIONALLY HIGH TEMP. ',
     .          'AND ANOMALOUS CONSEQUENCE MAY RESULT.',/,' CHECK ',
     .          '"QCENT" AND "HEAT" IN THE INPUT MAY BE NEEDED.')
880  FORMAT (/,' * ERROR *  FUME CONCENTRATION (CONC) HIGHER THAN ',
     .          '100% (CONC=',F5.0,')',/,' FOR FIRE SOURCE',I5)
890  FORMAT (/,1X,'* WARNING *  (HTPO2=',F5.0,') FOR FIRE SOURCE',I5,
     .          ' IS UNLIKELY HIGH,',/,' ANOMALOUS CONSEQUENCE MAY ',
     .          'RESULT.')
900  FORMAT (/,1X,' * WARNING *  AIRWAY OR STARTING JUNCTION NUMBER ',
     .          'OF FIRE',/,1X,'SOURCE ',I5,' IS NOT ON THE LIST. DATA ',
     .          'IGNORED.')
910  FORMAT (//,1X,71('*'),///,22X,'OUTPUT OF PROGRAM MFIRE',///,1X,
     .          71('*'))
920  FORMAT (/,1X,'* ERROR *  NEGATIVE QF INPUT DETECTED FOR FAN',I5)
C
     END
C
C
     SUBROUTINE CHECK1 (NSTOP,MAXNO,KV)
C
```

```
C     ------------------------------------------------------------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       1) MAKE-UP OF THE MISSING DATA.
C       2) CHECK FOR CONSISTENCY OF INPUT DATA.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
        DO 10 I=1,NB
           IF (NO(I).GT.MAXNO) MAXNO=NO(I)
           IF (NWTYP(I).EQ.0) THEN
              IF (R(I).LE.0.0) THEN
                 IF (A(I).LE.0.0.OR.LA(I).LE.0.0) THEN
                    WRITE (8,130) NO(I)
                    KV=NB
                    NSTOP=1
                    RETURN
                 ELSE
                    R(I)=KF(I)*LA(I)*O(I)/(5.2*A(I)**3)*DR/0.075
                 ENDIF
              ENDIF
           ENDIF
           RSTD(I)=R(I)
10      CONTINUE
C
C
        MAXJ=0
        DO 30 I=1,NB
           IF (JS(I).EQ.JF(I)) THEN
              WRITE (8,100) NO(I)
              KV=NB
              NSTOP=1
              RETURN
           ENDIF
           NREV(I)=0
           N1=0
           N2=0
           IF (NVPN.LE.0) THEN
              KV=NJ
              DO 20 J=1,NJ
                 IF (JNO(J).GT.MAXJ.AND.I.EQ.1) MAXJ=JNO(J)
                 IF (JS(I).EQ.JNO(J)) N1=N1+1
                 IF (JF(I).EQ.JNO(J)) N2=N2+1
20            CONTINUE
              IF (N1.EQ.0) THEN
                 WRITE (8,105) NO(I),JS(I)
                 NSTOP=2
              ELSE IF (N1.GE.2) THEN
                 WRITE (8,110) JS(I)
                 NSTOP=2
              ENDIF
              IF (N2.EQ.0) THEN
```

```
40
                WRITE (8,105) NO(I),JF(I)
                NSTOP=2
             ELSE IF (N2.GE.2) THEN
                WRITE (8,110) JF(I)
                NSTOP=2
             ENDIF
             IF (NSTOP.GT.0) RETURN
          ENDIF
30     CONTINUE
       IF (NVPN.LE.0) THEN
          KV=NJ
          DO 50 J=1,NJ
             J1=0
             DO 40 I=1,NB
                IF (JS(I).EQ.JNO(J).OR.JF(I).EQ.JNO(J)) THEN
                   J1=J1+1
                ENDIF
40           CONTINUE
             IF (J1.EQ.0) THEN
                WRITE (8,115) JNO(J)
                NSTOP=2
                RETURN
             ELSE IF (J1.EQ.1) THEN
                WRITE (8,160) JNO(J)
                NSTOP=2
                RETURN
             ENDIF
50        CONTINUE
       ENDIF
       DO 70 I=1,NB-1
          J1=0
          DO 60 J=I+1,NB
             IF (NO(I).EQ.NO(J)) J1=J1+1
60        CONTINUE
          IF (J1.GT.0) THEN
             WRITE (8,140) NO(I)
             KV=NB
             NSTOP=1
             RETURN
          ENDIF
70     CONTINUE
       IF (NVPN.LE.0) THEN
          DO 90 I=1,NJ-1
             J2=0
             DO 80 J=I+1,NJ
                IF (JNO(I).EQ.JNO(J)) J2=J2+1
80           CONTINUE
             IF (J2.GE.1) THEN
                WRITE (8,150) JNO(I)
                KV=NJ
                NSTOP=2
                RETURN
             ENDIF
90        CONTINUE
```

```
      ENDIF
C
100   FORMAT (/,1X,'* ERROR *  AIRWAY',I5,' IS ISOLATED FROM ',
     .        'THE NETWORK.')
105   FORMAT (/,1X,'* ERROR *  AIRWAY',I5,' IS ISOLATED FROM ',
     .        'THE NETWORK ',/,' FOR THE UNSPECIFIED END',I5,' IN ',
     .        'JNO LIST.')
110   FORMAT (/,1X,'* ERROR *  DUPLICATED JUNCTION NUMBER',I5,' IN',
     .        ' JNO LIST.')
115   FORMAT (/,1X,'* ERROR *  JUNCTION',I4,' IS ISOLATED FROM ',
     .        'THE NETWORK.')
130   FORMAT (/,1X,'* ERROR *  NO RESISTANCE AND DIMENSION WERE',
     .        ' STATED FOR AIRWAY',I5)
140   FORMAT (/,1X,'* ERROR *  DUPLICATED AIRWAY ID NUMBER',I5,
     .        ' IN THE AIRWAY CARDS.')
150   FORMAT (/,1X,'* ERROR *  DUPLICATED JUNCTION ID NUMBER',I5,
     .        ' IN THE JUNCTION CARDS.')
160   FORMAT (/,1X,'* ERROR *  JUNCTION ',I4,' IS A DEADEND.')
170   FORMAT (/,1X,'* ERROR *  AIRWAY',I5,' IS ISOLATED FROM THE ',
     .        'NETWORK',/,' DUE TO DEADEND',I6)
C
      RETURN
      END
C
C
      SUBROUTINE ARR (MARKX)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSES:
C        1) GROUPING AIRWAYS ACCORDING TO THEIR TYPES.
C        2) ARRANGING REGULAR AIRWAYS ACCORDING TO THE PRODUCT OF
C           Q AND R.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
C        ORDER OF AIRWAYS IN ARRAY INU: (1) FIXED-Q; (2) FAN; (3) HIGH-
C     RQ; (4) LOW-RQ; (5) FIXED-P.
C
      NBU=NB
      NBL=1
      DO 10 K=1,NB
         IF (NWTYP(K).LT.0) THEN
            INU(NBU)=K
            NBU=NBU-1
         ELSE IF (NWTYP(K).EQ.0.OR.NWTYP(K).EQ.10) THEN
            RQ(K)=ABS(R(K)*Q(K))
            NWTYP(K)=2
         ELSE
            IF (NFNUM.GT.0) THEN
               DO 5 J=1,NFNUM
                  IF (NOF(J).EQ.NO(K)) GO TO 10
```

```
5             CONTINUE
            ENDIF
            INU(NBL)=K
            NBL=NBL+1
          ENDIF
10    CONTINUE
      IF (NFNUM.GT.0) THEN
        DO 25 K=1,NB
          DO 20 J=1,NFNUM
            IF (NOF(J).EQ.NO(K)) THEN
              RGRAD(J)=0.0
              NFCW(J)=0
              NFREG(J)=K
              INU(NBU)=K
              NBU=NBU-1
              GO TO 25
            ENDIF
20        CONTINUE
25      CONTINUE
      ENDIF
C
C
      IF (NBU.GT.(NBL-1)) THEN
30      NRETU=0
        DO 40 K=1,NB
          IF (NWTYP(K).GE.2) THEN
            IF (NRETU.LE.0) THEN
              NMIN=K
              NRETU=1
            ENDIF
            IF (RQ(NMIN).GT.RQ(K)) NMIN=K
          ENDIF
40      CONTINUE
        INU(NBL)=NMIN
        NBL=NBL+1
        NWTYP(NMIN)=0
        IF ((NBU+1).GT.NBL) GO TO 30
      ENDIF
C
C       NWTYP(I)=10: AIRWAYS CONTAINING FIRE SOURCES.
C       MARKX=1: CALLED FROM DYNAMIC SIMULATION PART.
C
      IF (MARKX.GT.0) THEN
        DO 60 I=1,INFLOW
          DO 50 J=1,NB
            IF (NCENT(I).EQ.NO(J)) THEN
              NWTYP(J)=10
              GO TO 60
            ENDIF
50        CONTINUE
60      CONTINUE
      ENDIF
C
      RETURN
```

```
      END
C
C
      SUBROUTINE BASE
C
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSE:
C        FORMATION OF BASE SYSTEM.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
C
      IND=INU(1)
      KJF(NJ)=JS(IND)
      KJS(NJ-1)=JS(IND)
      JS(IND)=-JS(IND)
      KJF(NJ-1)=JF(IND)
      KNO(NJ-1)=IND
      KNUM=NJ-1
10    DO 30 NUC=2,NB
         IND=INU(NUC)
         IF (JS(IND).LT.0) GO TO 30
         N1=0
         N0=0
         DO 20 K=KNUM,NJ
            IF (JS(IND).EQ.KJF(K)) N0=1
            IF (JF(IND).EQ.KJF(K)) N1=1
20       CONTINUE
         IF ((N1+N0).EQ.1) THEN
           IF (NWTYP(IND).LT.0) WRITE (8,100) NO(IND)
           KNUM=KNUM-1
           IF (N0.LE.0) THEN
             KJS(KNUM)=JF(IND)
             KJF(KNUM)=JS(IND)
             KNO(KNUM)=-IND
             JS(IND)=-JS(IND)
           ELSE
             KJS(KNUM)=JS(IND)
             KJF(KNUM)=JF(IND)
             KNO(KNUM)=IND
             JS(IND)=-JS(IND)
           ENDIF
           GO TO 10
         ELSE IF ((N1+N0).GT.1) THEN
           JS(IND)=-JS(IND)
           JF(IND)=-JF(IND)
         ENDIF
30    CONTINUE
      DO 40 K=1,NB
         JS(K)=IABS(JS(K))
```

```
40    CONTINUE
C
100   FORMAT (/,1X,'*ATTENTION*   REGULATOR ',I5,' IN BASE SYST.')
C
      RETURN
      END
C
C
      SUBROUTINE MSLIST
C
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSES:
C        SET-UP OF MESH LIST.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
      MESC=0
      MNO=0
      DO 60 K=1,NB
         IF (JF(K).LT.0) THEN
           JF(K)=-JF(K)
           MNO=MNO+1
           JBM=JS(K)
           JEM=JF(K)
           NK=K
10         MESC=MESC+1
           MEND(MNO)=MESC
           MSL(MESC)=NK
C
C
           DO 20 KCO=1,NJ
              IF (JBM.EQ.KJF(KCO)) THEN
                KB=KCO
                GO TO 30
              ENDIF
20         CONTINUE
30         DO 40 KCO=1,NJ
              IF (JEM.EQ.KJF(KCO)) THEN
                KE=KCO
                GO TO 50
              ENDIF
40         CONTINUE
50         IF (KB.LT.KE) THEN
             NK=KNO(KB)
             JBM=KJS(KB)
             GO TO 10
           ELSE IF (KB.GT.KE) THEN
             NK=-KNO(KE)
             JEM=KJS(KE)
             GO TO 10
```

```
          ENDIF
        ENDIF
60    CONTINUE
      IF (MEND(MNO).GT.LMX) WRITE (8,100)
C
      RETURN
100   FORMAT (/,1X,'* ATTENTION *  CAPACITY OF MSL EXCEEDED, INCREASE',
     .            ' LMX.')


      END
C
C
      SUBROUTINE MBLNC
C
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       REALIZATION OF BALANCE OF MASS FLOW RATES.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
      MBEGW=2
      DO 20 K=1,MNO
          MENDW=MEND(K)
          DO 10 J=MBEGW,MENDW
              N=MSL(J)
              IF (N.LT.0) N=-N
              Q(N)=0.
10        CONTINUE
          MBEGW=MENDW+2
20    CONTINUE
C
C
      MBEGW=1
      DO 40 K=1,MNO
          MENDW=MEND(K)
          N=MSL(MBEGW)
          IF (N.LT.0) N=-N
          Q1=Q(N)
          M=MBEGW+1
          DO 30 J=M,MENDW
              N=MSL(J)
              FACT=1.
              IF (N.LT.0) THEN
              N=-N
              FACT=-1.
              ENDIF
              Q(N)=Q(N)+Q1*FACT
30        CONTINUE
          MBEGW=MENDW+1
```

```
40    CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE NVP1
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSE:
C        EVALUATION OF NVP THROUGH MANIPULATION OF INPUT JUNCTION DATA
C        IN TEMP. AND ELEVATION.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
      MBEGW=1
      DO 30 K=1,MNO
         FNVP(K)=0.
         IF (NVPN.GE.1) GO TO 30
         MENDW=MEND(K)
         NT=0
         TSU=0.
         DO 20 J=MBEGW,MENDW
            N=MSL(J)
            NX=N
            IF (NX.LT.0) NX=-NX
            DO 10 L=1,NJ
               IF (JS(NX).EQ.JNO(L)) THEN
                  T0=T(L)+460.
                  Z0=Z(L)
               ELSE IF (JF(NX).EQ.JNO(L)) THEN
                  T1=T(L)+460.
                  Z1=Z(L)
               ENDIF
10          CONTINUE
            H=T0*Z1-T1*Z0
            IF (N.LT.0) H=-H
            FNVP(K)=FNVP(K)+H
            TSU=TSU+T0+T1
            NT=NT+1
20       CONTINUE
         TM=TSU/NT
C
C
         FNVP(K)=FNVP(K)*DR/(5.2*TM)
         MBEGW=MENDW+1
30       CONTINUE
C
      RETURN
      END
C
```

```
C
      SUBROUTINE ITR (MARKX,NSFLOW,NSNVP,MADJC,ITCT)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSE:
C        APPLICATION OF HARDY CROSS METHOD TO NETWORK BALANCING.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
5     IT=0
10    DQSUM=0.
      MBEGW=1
      DO 50 K=1,MNO
         MENDW=MEND(K)
         DPSUM=0.
         RQSUM=0.
         N=MSL(MBEGW)
         IF (N.LT.0) N=-N
         IF (NWTYP(N).EQ.(-1)) THEN
            MBEGW=MENDW+1
            GO TO 50
         ENDIF
         DO 30 J=MBEGW,MENDW
            N=MSL(J)
            FACT=1.
            IF (N.LT.0) THEN
               N=-N
               FACT=-1.
            ENDIF
            IF(NWTYP(N).EQ.1) THEN
               IF (NFNUM.GT.0) THEN
                  DO 20 L=1,NFNUM
                     IF (NFREG(L).EQ.N) THEN
                        RQSUM=RQSUM-RGRAD(L)
                        GO TO 25
                     ENDIF
20                CONTINUE
               ENDIF
25             IF (IABS(JS(N)).NE.JSB(N)) FACT=-FACT
               DPSUM=DPSUM-FACT*R(N)
            ELSE
               RQ2=R(N)*ABS(Q(N))*2.E-5
               RQSUM=RQSUM+RQ2
               DP=R(N)*Q(N)*ABS(Q(N))*1.E-10
               DPSUM=DPSUM+FACT*DP
            ENDIF
30       CONTINUE
         DQ=(DPSUM-FNVP(K))*1.E5/RQSUM
         DO 40 J=MBEGW,MENDW
            N=MSL(J)
            FACT=1.
```

```
            IF (N.LT.0) THEN
               N=-N
               FACT=-1.
            ENDIF
            Q(N)=Q(N)-(DQ*FACT)
40       CONTINUE
         DQSUM=DQSUM+ABS(DQ)
         MBEGW=MENDW+1
50    CONTINUE
C
C
      DO 100 KI=1,NB
         IF (NWTYP(KI).EQ.1.AND.NFNUM.GT.0) THEN
            DO 90 J=1,NFNUM
               IF (NOF(J).EQ.NO(KI)) THEN
                  NFCW(J)=0
                  NEGQ(J)=0
                  IF (Q(KI).LT.0.0.AND.IABS(JS(KI)).EQ.JSB(KI))
     .               NEGQ(J)=1
                  IF (Q(KI).GT.0.0.AND.IABS(JS(KI)).NE.JSB(KI))
     .               NEGQ(J)=1
                  NPTS=MPTS(J)
                  IF (NVPN.GE.1.AND.MADJC.LE.0) THEN
                     TABF=TR
                  ELSE
                     NABF=JS(KI)
                     DO 70 L=1,NJ
                        IF (NABF.EQ.JNO(L)) THEN
                           TABF=T(L)
                           GO TO 80
                        ENDIF
70                   CONTINUE
                  ENDIF
80                IF (NSKP(J).EQ.0) THEN
                     CALL SPLINE (J,Q(KI),R(KI),TABF,0)
                     NSKP(J)=1
                  ELSE
                     CALL SPLINE (J,Q(KI),R(KI),TABF,1)
                  ENDIF
                  GO TO 100
               ENDIF
90          CONTINUE
         ENDIF
100   CONTINUE
C
C
      IT=IT+1
         ITCT=ITCT+1
         IF (IT.LE.1) THEN
            GO TO 10
         ELSE IF ((DQSUM/FLOAT(MNO)).LT.10.) THEN
            RETURN
         ELSE IF (ITCT.GT.100) THEN
            WRITE (8,200) DQSUM/FLOAT(MNO)
```

```
            RETURN
          ELSE IF (IT.GT.20) THEN
C
            NSFLOW=0
            CALL ARR (MARKX)
            CALL BASE
            CALL MSLIST
            CALL MBLNC
C
C     NSNVP: 0: SUB. IS CALLED FROM NETWORK PART.  1: SUB. IS
C              CALLED FROM TEMP. OR DYNAMIC SIMULATION PART.
C
            IF (NSNVP.LE.0) THEN
              IF (NVPN.LE.0) CALL NVP1
            ELSE
              CALL NVP2
            ENDIF
            GO TO 5
          ENDIF
      GO TO 10
C
200   FORMAT (//,1X,'* ATTENTION *  ACCURACY CRITERION IN NETWORK',
     .          ' BALANCING',/,1X,'( SUB. ITR ) WAS NOT SATISFIED IN 100'
     .          ,'ITERATIONS. THE',/,' CRITERION: 10 PER MESH,',6X,
     .          ' CURRENT ERROR:',F8.0)
210   FORMAT (//,1X,'* ATTENTION *  ACCURACY CRITERION IN NETWORK',
     .          ' BALANCING',/,1X,'( SUB. ITR ) WAS NOT SATISFIED IN 100'
     .          ,'ITERATIONS. THE',/,' CRITERION: 10 PER MESH,',6X,
     .          ' CURRENT ERROR:',F8.0,/,' IT WAS HAPPENED IN DATA ',
     .          'PREPARATION CYCLE ',I5)
C
      END
C
C
      SUBROUTINE SPLINE (J,QKI,RKI,TABF,NTW)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       1) FAN CURVE FITTING.
C       2) FAN PRESSURE INTERPOLATION.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
C     FKQ(J,I,1-4)COEFFICIENTS OF 3RD ORDER POLYNOMIAL FOR FAN NOF(J),
C                 SEGMENT I.
C     NSKP(J)=1: MARKER INDICATING THAT FAN CURVE FITTING FOR FAN
C                NOF(J) HAS BEEN PROCESSED.
C
      MF = MPTS(J)
      IF (NTW.EQ.0) THEN
        DELQ=1.0
```

```
50
      DELP=0.0
      DQP=0.0
      MFA = MF-1
      DO 10 I = 1,MFA
          FK(I) = 0.0
          DK(I) = 0.0
          DO 10 K = 1,MFA
              FC(I,K) = 0.0
10    CONTINUE
      FC(1,1) = 2.0
      FC(1,2) = 1.0
      FC(MFA,MFA-1) = 1.0
      FC(MFA,MFA) = 2.0
      DO 20 I = 1,MFA
          DQA = DELQ
          DELQ = QF(J,I+1)-QF(J,I)
          DPA = DELP
          DELP = PF(J,I+1)-PF(J,I)
          DQPA = DQP
          DQP = DELP/DELQ
          IF (I.GT.1.AND.I.LT.MFA) THEN
             DK(I) = (DQPA*DELQ+DQP*DQA)*3.0
             FC(I,I-1) = DELQ
             FC(I,I) = 2.0*(DQA+DELQ)
             FC(I,I+1) = DQA
          ELSE
             DK(I) = DQP*3.0
          ENDIF
20    CONTINUE
      DO 25 I = 2,MFA
          FC(I,I) = FC(I,I)-FC(I-1,I)*FC(I,I-1)/FC(I-1,I-1)
          DK(I) = DK(I)-DK(I-1)*FC(I,I-1)/FC(I-1,I-1)
25    CONTINUE
      FK(MFA) = DK(MFA)/FC(MFA,MFA)
C
C
      DO 30 I = 2,MFA
          II = MFA-I+1
          FK(II) = (DK(II)-FC(II,II+1)*FK(II+1))/FC(II,II)
30    CONTINUE
      DO 35 I = 1,MFA
          FXD = QF(J,I+1)-QF(J,I)
          FXA = 1.0/FXD**2.
          FXB = FK(I)-(PF(J,I+1)-PF(J,I))/FXD
          FXC = FK(I+1)-(PF(J,I+1)-PF(J,I))/FXD
          FKQ(J,I,1) = (QF(J,I+1)*PF(J,I)-QF(J,I)*PF(J,I+1))
     .                 /FXD-FXA*(FXB*QF(J,I)*QF(J,I+1)**2+
     .                 FXC*QF(J,I+1)*QF(J,I)**2)
          FKQ(J,I,2) = (PF(J,I+1)-PF(J,I))/FXD+FXA*(FXB*
     .                 (QF(J,I+1)**2+2.*QF(J,I+1)*QF(J,I))+
     .                 FXC*(QF(J,I)**2+2.*QF(J,I+1)*QF(J,I)))
          FKQ(J,I,3) = -FXA*(FXB*(2.*QF(J,I+1)+QF(J,I))+FXC*
     .                 (QF(J,I+1)+2.*QF(J,I)))
          FKQ(J,I,4) = FXA*(FXB+FXC)
```

```
35       CONTINUE
         NSKP(J)=1
       ENDIF
C
C
       QKI=ABS(QKI)
       IF (QKI.LE.QF(J,1).OR.NEGQ(J).GE.1) THEN
         RKI = PF(J,1)*(TR+460.)/(TABF+460.)
         RGRAD(J) = 0.0
         NFCW(J) = 1
         RETURN
       ELSE IF (QKI.GE.QF(J,MF)) THEN
         RKI = PF(J,MF)*(TR+460.)/(TABF+460.)
         RGRAD(J) = 0.0
         NFCW(J) = 1
         RETURN
       ENDIF
       IF (QF(J,1).GE.QKI) THEN
         M = 1
       ELSE
         DO 40 K = 2,MF
           IF (QF(J,K).GE.QKI) THEN
             M = K-1
             GO TO 45
           ENDIF
40       CONTINUE
       ENDIF
       M = MFA
45     RKI = FKQ(J,M,1)+FKQ(J,M,2)*QKI+FKQ(J,M,3)*QKI**2
      .          +FKQ(J,M,4)*QKI**3
       RKI=RKI*(TR+460.)/(TABF+460.)
       RGRAD(J) = FKQ(J,M,2)+2*FKQ(J,M,3)*QKI+
      .                3*FKQ(J,M,4)*QKI**2
C
       RETURN
       END

C
C
       SUBROUTINE RGLT (MARKY)
C
C      ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSE:
C        CALCULATION OF VENTILATION RESISTANCES OF REGULATORS REQUIRED
C      BY THE USER-SPECIFIED AIRFLOW RATES IN THE AIRWAYS.
C
C      ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
       MBEGW=1
       DO 20 K=1,MNO
         DPSUM=0.
```

```
            MENDW=MEND(K)
            NX=MSL(MBEGW)
            IF (NX.LT.0) NX=-NX
            IF (NWTYP(NX).LT.0) THEN
               M=MBEGW+1
               DO 10 J=M,MENDW
                  IF (MSL(J).GT.0) THEN
                     N=MSL(J)
                     FACT=1.
                  ELSE
                     N=-MSL(J)
                     FACT=-1.
                  ENDIF
                  IF (NWTYP(N).GT.0) THEN
                     DP=-R(N)
                  ELSE
                     DP=R(N)*Q(N)*ABS(Q(N))*1.E-10
                  ENDIF
                  DPSUM=DPSUM+DP*FACT
10             CONTINUE
               IF (NVPN.LE.0) DPSUM=DPSUM-FNVP(K)
               RSTD(NX) =-DPSUM*1.E10/(Q(NX))**2
               R(NX)=RSTD(NX)
               IF (R(NX).LT.0.0) THEN
                  R(NX)=DPSUM
                  RSTD(NX)=R(NX)
                  NWTYP(NX)=1
                  IF (MARKY.LE.0) WRITE (8,30) R(NX),NO(NX)
               ELSE
                  NWTYP(NX)=0
               ENDIF
            ENDIF
            MBEGW=MENDW+1
20       CONTINUE
C
30       FORMAT (//,1X,'* ATTENTION *  A CONSTANT PRESSURE FAN OF ',F6.3,
     .             ' IN.W.G.',/,1X,'IS INSTALLED IN PREVIOUSLY FIXED-Q ',
     .             'AIRWAY ',I5)
C
         RETURN
         END
C
C
         SUBROUTINE CCDATA (MARKY,NSTOP)
C
C        ------------------------------------------------------------------------------------------------------------------------
C
C           SUBROUTINE PURPOSES:
C           1) COMPLETION OF INPUT DATA.
C           2) EVALUATION OF ROCK TEMP. AND AIRWAY WETNESS.
C
C        ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
```

```
C
      DIMENSION DAL(15),NTT(NMX),LA1(NMX)
      DOUBLE PRECISION X,LA1
C
C
      HAAVR=-1.0
      HKAVR=-1.0
      KFAVR=-1
      LAAVR=-1
      AAVR=-1.0
      OAVR=-1.0
      IF (NAV.GT.0) THEN
10       CALL READIN (DAL,6,ISTOP,0)
         IF (ISTOP.EQ.1) THEN
            WRITE (8,240)
            WRITE (8,250) (ROW(IE),IE=1,80)
            NSTOP=5
            RETURN
         ENDIF
         IF (MARKY.EQ.0) WRITE (6,260)
         IF (DAL(15).LT.(-1.E20)) GO TO 10
         IF (NCOMTS.GT.NCOMT2) THEN
            NCOMT2=NCOMTS
            GO TO 10
         ENDIF
         IF (DAL(1).LT.0.0.OR.DAL(2).LT.0.0.OR.DAL(3).LT.0.0.OR.
     .        DAL(4).LT.0.0.OR.DAL(5).LT.0.0.OR.DAL(6).LT.0.0) THEN
            WRITE (8,200)
            WRITE (8,240)
            WRITE (8,250) (ROW(IE),IE=1,80)
            NSTOP=5
            RETURN
         ENDIF
         HAAVR=DAL(1)
         HKAVR=DAL(2)
         KFAVR=DAL(3)
         LAAVR=DAL(4)
         AAVR=DAL(5)
         OAVR=DAL(6)
      ENDIF
      IF (HAAVR.LT.1.E-5.OR.HAAVR.GT.10.0) HAAVR=0.1
      IF (HKAVR.LT.1.E-5.OR.HKAVR.GT.50.0) HKAVR=3.0
      IF (KFAVR.LT.10.OR.KFAVR.GT.1000) KFAVR=100
      IF (LAAVR.LT.10.OR.LAAVR.GT.3000) LAAVR=500
      IF (AAVR.LT.0.1.OR.AAVR.GT.500.0) AAVR=100.0
      IF (OAVR.LT.0.1.OR.OAVR.GT.100.0) OAVR=40.0
      IF (MARKY.LE.0) THEN
         IF (NAV.LE.0.AND.IOUT.LT.0) WRITE (8,220)
         WRITE (8,230) HAAVR,HKAVR,KFAVR,LAAVR,AAVR,OAVR
      ENDIF
C
C
      DO 50 I=1,NB
          IF (HA(I).LE.1.E-20.OR.HA(I).GT.50.0) HA(I)=HAAVR
```

```
      IF (HK(I).LE.1.E-20.OR.HK(I).GT.100.0) HK(I)=HKAVR
      IF (KF(I).LE.0.OR.KF(I).GT.2000.0) KF(I)=KFAVR
      IF (LA(I).LE.1.E-5.OR.LA(I).GT.1.E5) LA(I)=FLOAT(LAAVR)
      IF (A(I).LE.1.E-3.OR.A(I).GT.1000.0) A(I)=AAVR
      IF (O(I).LE.1.E-3.OR.O(I).GT.1000.0) O(I)=OAVR
50  CONTINUE
C
C
    IIT=0
    DO 80 J=1,NB
      IF (NO(J).LE.0) NO(J)=-NO(J)
      L=0
      M=0
      DO 60 I=1,NJ
        IF (JS(J).EQ.JNO(I)) THEN
          ES=Z(I)
          TRS=T(I)
          L=1
        ELSE IF (JF(J).EQ.JNO(I)) THEN
          EF=Z(I)
          TRF=T(I)
          M=1
        ENDIF
        IF ((L+M).EQ.2) THEN
          DZRD(J)=EF-ES
          IF (LA(J).LT.ABS(DZRD(J))) THEN
            LA(J)=ABS(DZRD(J))
            IF (MARKY.LE.0) WRITE (8,270) NO(J),LA(J)
            IF (LA(J).GT.5000..AND.MARKY.LE.0) WRITE(8,280) NO(J)
          ENDIF
          IF (JF(J).EQ.JSTART) TRF=TRS
          IF (TROCK(J).GT.150.0.AND.MARKY.LE.0)
     .      WRITE (8,300) TROCK(J),NO(J)
          IF (TROCK(J).LT.1.E-3.OR.TROCK(J).GT.150.0) THEN
            X=0.014*O(J)/(A(J)**0.8*(ABS(Q(J)))**0.2)
            LA1(J)=LA(J)
            XYZ=DEXP(-X*LA1(J))
            TROCK(J)=(TRF-TRS*XYZ+DZRD(J)*
     .          (1.+XYZ)/(2*187.))/(1.0-XYZ)
          ENDIF
          GO TO 70
        ENDIF
60    CONTINUE
70    IF (TROCK(J).GT.150.0.OR.TROCK(J).LT.0.0) THEN
        IIT=IIT+1
        NTT(IIT)=NO(J)
        TROCK(J)=TRF
      ENDIF
80  CONTINUE
    IF (IIT.GT.0.AND.(MARKY.LE.0.OR.IOUT.LE.(-2))) THEN
      WRITE (8,290) (NTT(JJ),JJ=1,IIT)
      WRITE (8,295)
    ENDIF
    RETURN
```

```
C
200  FORMAT (/,1X,'* ERROR *  ILLEGAL NEGATIVE VALUE IN AVE.-VALUE',
     .          ' CARD DETECTED.')
220  FORMAT (//,1X,'AVE.-VALUE CARD WAS OMITTED BY USER (NAV=0)')
230  FORMAT (///,1X,'THE FOLLOWING AVERAGE VALUES WILL BE USED TO ',
     .          'FILL ANY MISSING DATA:',//,1X,'THERMAL DIFFUSIVITY OF',
     .          ' ROCK',T30,F9.3,T43,'FT2/HR',/,1X,'ROCK THERMAL ',
     .          'CONDUCTIVITY',T30,F9.3,T43,'BTU/(HR*FT*F)',/,1X,
     .          'FRICTION FACTOR',T30,I9,T43,'1.E-10 LBF*MIN2/FT4',/,1X,
     .          'AIRWAY LENGTH',T30,I9,T43,'FT',/,1X,'SECTIONAL AREA ',
     .          T30,F9.3,T43,'FT2',/,1X,'PERIMETER',T30,F9.3,T43,'FT',/)
240  FORMAT (/,' THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .          'IN:',//,' HAAVR,HKAVR,KFAVR,LAAVR,AAVR,OAVR')
250  FORMAT (/,' THE CURRENT INPUT WAS:',//,80A1)
260  FORMAT (/,5X,'AVERAGE-VALUE CARD WAS READ IN.')
270  FORMAT (/,1X,'* ATTENTION *  THE LENGTH OF AIRWAY',I5,' WAS LESS'
     .          ,' THAN ITS ELEVATION',/,' DIFFERENCE. THEY ARE NOW ',
     .          'TAKEN EQUAL TO EACH OTHER AS: ',F9.1,' FT')
280  FORMAT (/,1X,'* WARNING *  THE ELEVATION DIFFERENCE OF AIRWAY',
     .          I5,' IS UNLIKELY LARGE.')
290  FORMAT (/,1X,'* ATTENTION *  UNLIKELY VALUES OF ROCK TEMPERATURE',
     .          ' WERE OBTAINED IN THE',/,' FOLLOWING  AIRWAY(S) BASED ON',
     .          ' USER SPECIFIED HEAT TRANSFER PARAMETERS.',/,10(12I6,/))
295  FORMAT (' THEY WERE MODIFIED AND SHOWN IN FOLLOWING TABLE(S).')
300  FORMAT (/,1X,'* ATTENTION *  THE UNLIKELY INPUT VALUE OF ROCK ',
     .          ' TEMPERATURE (',F6.0,' F)',/,' FOR AIRWAY',I5,' IS ',
     .          'IGNORED.')
C
      END
C
C
      SUBROUTINE  FWCT (NSFLOW)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSES:
C        1) DATA REARRANGEMENT FOR AIRWAYS IN WHICH AIRFLOW REVERSAL
C           HAPPENED.
C        2) FORMATION OF AIRFLOW SCHEME IN THE NETWORK.
C        3) RELATING JNO AND JNOL LISTS.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
      DO 20 I=1,NB
         IF (Q(I).LT.0.0) THEN
            NSFLOW=0
            J=JS(I)
            JS(I)=JF(I)
            JF(I)=J
            Q(I)=-Q(I)
            DZRD(I)=-DZRD(I)
C
```

```
C     NNREV(K): NUMBER OF AIRFLOW DIRECTION CHANGES IN AIRWAY NO(K).
C
                IF (NWTYP(I).EQ.1) R(I)=-R(I)
                NNREV(I)=NNREV(I)+1
            ENDIF
20    CONTINUE
C
C
      L=0
      M=0
      N=1
      DO 60 I=1,MAXJ
          K=L
          DO 40 J=1,NB
              IF (JS(J).EQ.I) THEN
                  L=L+1
                  NGOUT(L)=J
              ENDIF
40        CONTINUE
          LOUT(N)=L
          MM=M
          DO 50 J=1,NB
              IF (JF(J).EQ.I) THEN
                  M=M+1
                  NGIN(M)=J
              ENDIF
50        CONTINUE
          MMIN(N)=M
          IF ((MM.NE.M).OR.(K.NE.L)) THEN
             JNOL(N)=I
             N=N+1
          ENDIF
60    CONTINUE
      DO 80 I=1,NJ
C
C     JNOL(J): LIST OF JUNCTION NUMBERS IN INCREASING ORDER.
C     JLR(J): BRIDGE ARRAY RELATING JNO AND JNOL LISTS.
C
          DO 70 J=1,NJ
              IF (JNOL(I).EQ.JNO(J)) THEN
                  JLR(I)=J
                  GO TO 80
              ENDIF
70        CONTINUE
80    CONTINUE
      RETURN
C
      END
C
C
      SUBROUTINE  CH4EVA
C
C     ------------------------------------------------------------------------------------------------------------------------
C
```

```
C       SUBROUTINE PURPOSE:
C       EVALUATION OF METHANE EMISSION RATES IN AIRWAYS.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
C     CH4C(J): CH4 CONCENTRATION IN JUNCTION JNO(J).
C     CH4V(I): CH4 EMISSION RATES IN AIRWAY NO(I).
C     CH4PA(I): CH4 EMISSION RATE PER UNIT SURFACE AREA.
C
      DO 40 I=1,NB
         IF (CH4V(I).GT.0.0) GO TO 40
         CH4V(I)=CH4PA(I)*LA(I)*O(I)
         IF (CH4V(I).GT.0.0) GO TO 40
         M=0
         N=0
         DO 30 L=1,NJ
            IF (JS(I).EQ.JNO(L)) THEN
               CH4S=CH4C(L)
               M=1
            ENDIF
            IF (JF(I).EQ.JNO(L)) THEN
               CH4F=CH4C(L)
               N=1
            ENDIF
            IF (M+N.GT.1) THEN
               IF (CH4F.GT.CH4S) THEN
                  CH4V(I)=(CH4F-CH4S)*Q(I)/100.0
               ELSE
                  CH4V(I)=0.0
               ENDIF
               GO TO 40
            ENDIF
30       CONTINUE
40    CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE  CDENDS (MARKY,NSTART,L)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       EVALUATION OF AIR TEMP. AND CH4/FUME CONTENT AT AIRWAY ENDING.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
      PI=3.1415936
      PROPJS=PROP(L)
```

```
      JX=L
      CH4JS=PRCH4(L)
      JNO(L)=-JNO(L)
      JY=1
      IF (NSTART.NE.1) JY=LOUT(NSTART-1)+1
      JZ=LOUT(NSTART)
      DO 60 K=JY,JZ
         ICFTM=0
         I=NGOUT(K)
         NM=0
         CONTAM=0.0
         CONTQ=0.0
         HEATAD=0.0
         TFS=0.0
C
         IF (MARKY.EQ.0.AND.NTEMP.NE.1) GO TO 30
            DO 20 N=1,INFLOW
               IF (NCENT(N).EQ.NO(I)) THEN
                  NM=1
                  FACT=1.0
                  IF (QCENT(N).GT.10.0) THEN
                     Q1=Q(I)
                     IF (Q1.LT.QCENT(N)) Q1=Q1+QCENT(N)
                     FACT=1.0+(Q(I)-QCENT(N))/Q1
                  ENDIF
                  CONTAM=CONT(N)*CONC(N)/100.
                  CONTQ=CONT(N)
                  O2BEH(N)=(0.21-PROPJS)*100.0
                  HEATAD=HEAT(N)*FACT
C
                  IF (O2MIN(N).GT.0.0) THEN
                     O2SCL=O2MIN(N)+(21.0-O2MIN(N))*(FACT-1.0)
                     CONTAM=(0.21-PROPJS-O2SCL/100.)*Q(I)
                     O2SCLA=(0.21-PROPJS)*100.0
                     O2BEH(N)=MIN(O2SCL,O2SCLA)
                     IF (CONTAM.GT.0.0) THEN
                        HEATAD=CONTAM*437.
                     ELSE
                        HEATAD=0.0
                     ENDIF
                  ENDIF
                  IF (SMPO2(N).LE.0.0) GO TO 30
C
                  CONTAM=(0.21-PROPJS)*Q(I)*SMPO2(N)
                  O2BEH(N)=0.0
                  IF (FACT.LE.1.0) THEN
                     HTSCL=HTPO2(N)
                  ELSE
                     HTSCL=HTPO2(N)/FACT
                  ENDIF
                  IF (CONTAM.GT.0.0) THEN
                     HEATAD=(0.21-PROPJS)*Q(I)*HTSCL
                  ELSE
                     HEATAD=0.0
```

```
                ENDIF
                GO TO 30
              ENDIF
20          CONTINUE
C
C
30        RDCH4(I)=(CH4JS*Q(I)+CH4V(I))/(Q(I)+CH4V(I))
          CONTA=CONTAM
          IF (CONTA.LT.0.0) CONTA=0.0
          RDPROP(I)=(PROPJS*Q(I)+CONTA)/(Q(I)+CONTQ)
          TJS(I)=T(JX)
          IF (NM.GT.0) THEN
            HTAD(N)=HEATAD
            IF (HEATAD.NE.0.0) THEN
              VART=(9900.0+T(JX))**2+2.0*HEATAD/(Q(I)*DR*2.4E-5)
              TFS=-9900.0+SQRT(VART)
              TJS(I)=TFS
              TFSI(N)=TFS
            ELSE
              TFSI(N)=TJS(I)
            ENDIF
          ENDIF
C
C       AN ITERATION PROCESS IS EMPLOYED FOR MEAN AIR TEMPERATURE.
C     TM: MEAN AIR TEMP. IN AIRWAY.
C
          IF (ICFTM.GT.0) GO TO 40
          TM=(TJS(I)+TROCK(I))/2.
          GO TO 50
40        ARGMT=(TJS(I)-TROCK(I))/(TRD(I)-TROCK(I))
          IF (ARGMT.LE.1.) THEN
            WRITE (8,*) 'ERROR IN SUB CDENDS'
            TM=(TJS(I)+TRD(I))/2.
          ELSE
            X=ALOG(ARGMT)/LA(I)
            TM=TROCK(I)+(TJS(I)-TROCK(I))*(1.0-EXP(-X*LA(I)))/(X*LA(I))
          ENDIF
50        CALL KALPHA (OMIT,TM,FX,CP,I)
C
C     COAGE: COEFFICIENT OF AGE.
C     XNEW(I): EXPONENTIAL FACTOR IN COMPUTING AIR TEMP. VARIATION
C                ALONG AIRWAY NO(I).
C
          COAGE=BI(I)-FX*BI(I)**2/(0.375+BI(I))
          DCOAGE(I)=COAGE
          XNEW(I)=HK(I)*O(I)**2.*COAGE/(120.*DR*Q(I)*A(I))
          G=XNEW(I)*LA(I)/CP
          IF (G.GE.25.0) THEN
            GX=0.0
          ELSE
            GX=EXP(-G)
          ENDIF
          IF (ICFTM.LE.0) THEN
            TOLD=TROCK(I)
```

```
          ELSE
             TOLD=TRD(I)
          ENDIF
          TRD(I)=TROCK(I)+(TJS(I)-TROCK(I))*GX-
                    (DZRD(I)/(2.*778.26*CP))*(1.+GX)
          TDM=ABS(TOLD-TRD(I))
          IF (TDM.GT.50.) THEN
             ICFTM=1
             GO TO 40
          ENDIF
          IF (JF(I).GT.0) JF(I)=-JF(I)
60    CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE  KALPHA (TMST,TM,FX,CP,I)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSE:
C        EVALUATION OF "COEFFICIENT OF AGE" ASSUMING DRY CASE.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
      PI=3.1415936
      CP=0.24
      TMX=TM
      QX=Q(I)
      TX=TIME
C
C     WT: TEMP. CORRECTED AIR DENSITY, LBM/FT**3.
C     VISC: KINETIC VISCOSITY OF AIR, FT/HR.
C     HKA: THERMAL CONDUCTIVITY OF ROCK, BTU/HR*FT*F.
C     FO: FOURIER NUMBER.  BI: BIOT NUMBER.  RN: REYNOLDS NUMBER.
C
      WT=DR*(TR+460.)/(TMX+460.)
      VISC=0.000145*((460.+TMX)/492.)**1.75
      HKA=0.014*((460.+TMX)/492.)**0.81
      RN=QX*DR/(15.*WT*O(I)*VISC)
      FR0=0.0032+0.221/RN**0.237
      POT=(100./RN)**0.125
      COR=((KF(I)*0.075)/(809*DR*FR0))**POT
      HC=0.005*HKA*O(I)*RN**0.8*COR/A(I)
      FO=TX*HA(I)*O(I)**2/(4.*A(I)**2)
      BI(I)=HC*2.*A(I)/(O(I)*HK(I))
      X=(0.375+BI(I))*SQRT(FO)
C
C
      IF (X.LT.2.5) THEN
         N=0
```

```
      SUMT=0.0
      ADDT=X
10    SUMT=SUMT+ADDT
      N=N+1
      ADDT=-ADDT*X**2*(2.*N-1)/(N*(2.*N+1))
      IF (ABS(ADDT).GE.0.00001) GO TO 10
      FX=1.0-(EXP(X**2))*(1.0-(2.0/SQRT(PI))*SUMT)
    ELSE
      N=0
      SUMT=0.0
      ADDT=1.0
20    SUMT=SUMT+ADDT
      N=N+1
      OLADDT=ADDT
      ADDT=-ADDT*(2*N-1)/(2.0*X**2)
      IF (ABS(ADDT).LT.0.00001) GO TO 30
      IF (ABS(OLADDT).GT.ABS(ADDT)) GO TO 20
30    FX=1.0-1.0*SUMT/(SQRT(PI)*X)
    ENDIF
C
    RETURN
    END
C
C
    SUBROUTINE  CDJUNC (NSTART,JUMPBK,LL,KS)
C
C   ------------------------------------------------------------------------------------------------------------------------
C
C     SUBROUTINE PURPOSE:
C     EVALUATION OF AIR TEMP. AND CONTENT OF CONTAMINANTS IN JUNCTIONS
C
C   ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
C   KS=0: THE SUB. IS CALLED FOR FROM THE MAIN PROGRAM.
C   KS=1: THE SUB. IS CALLED FOR FROM SUB. RECIRC.
C
    DO 20 I=1,NJ
      L=JLR(I)
      IF (JNO(L).LT.0) GO TO 20
      IF (KS.EQ.1.AND.JNO(LL).NE.JNOL(I)) GO TO 20
      JY=1
      IF (I.GT.1) JY=MMIN(I-1)+1
      JZ=MMIN(I)
      SUMAIR=0.0
      SUMPR=0.0
      SUMCH4=0.0
      SUMHT=0.0
      DO 10 K=JY,JZ
        J=NGIN(K)
        IF (JF(J).GE.0.AND.KS.EQ.0) GO TO 20
        SUMAIR=SUMAIR+Q(J)
        SUMPR=SUMPR+RDPROP(J)*Q(J)
```

```
            SUMCH4=SUMCH4+RDCH4(J)*Q(J)
            SUMHT=SUMHT+TRD(J)*Q(J)*(0.2376+1.2E-5*TRD(J))
10        CONTINUE
C
C       IF DATA PREPARATION FOR ANY ONE OF THE JUNCTIONS IN THE
C     SYSTEM IS NOT COMPLETE, THE PROCESS OF DATA PREPARATION WILL
C     CONTINUE (JUMPBK=1).
C
          NSTART=I
          PROP(L)=SUMPR/SUMAIR
          PRCH4(L)=SUMCH4/SUMAIR
          RTCONT=9.801E7+2.0*SUMHT/(SUMAIR*2.4E-5)
          T(L)=-9900.+SQRT(RTCONT)
          LL=L
          IF (JAN.LE.1) GO TO 15
          DO 12 II=1,JAJ
              IF (L.EQ.JSTAR(II) ) THEN
                 T(L)=TSTAR(II)
                 PROP(L)=0.0
                 PRCH4(L)=0.0
              ENDIF
12        CONTINUE
15        JUMPBK=1
          RETURN
20    CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE  RECIRC (MRC,JUMPBK,NSTART,L)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       EVALUATION OF AIR TEMP., CONTAMINANT CONTENT AT AIRWAY ENDINGS
C     UNDER RECIRCULATION CONDITIONS.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
      NREC=1
10    LJ=0
      DO 60 I=1,NJ
          IF (JNO(I).LE.0) THEN
C
C         JUNCTION JNO(I) IS COMPLETED WITH THE NECESSARY CALCULATIONS
C     IF JNO(I)<0.
C
              LJ=LJ+1
          ELSE
              1N=0
              M=0
```

```
            SRPR=0.0
            SRCH4=0.0
            STRD=0.0
            QIN=0.0
            QREC=0.0
            DO 20 J=1,NB
               IF (JNO(I).EQ.(-JF(J))) THEN
                  N=N+1
                  SRPR=SRPR+RDPROP(J)*Q(J)
                  SRCH4=SRCH4+RDCH4(J)
                  STRD=STRD+TRD(J)
                  QIN=QIN+Q(J)
               ELSE IF (JNO(I).EQJF(J)) THEN
C
C      JF(J)>0 INDICATES THAT THE CALCULATION FOR AIRWAY NO(J) HAS
C     NOT BEEN COMPLETED.
C     MEMREC: TEMPORARY LIST OF AIRWAYS CARRYING RECIRC. AIR.
C
                  M=M+1
                  MEMREC(M)=J
                  QREC=QREC+Q(J)
               ENDIF
20          CONTINUE
            IF (N.LE.0) GO TO 60
            AVRPR=SRPR/QIN
            AVRCH4=SRCH4/N
            AVTRD=STRD/N
C
C      NREC SERVES AS FILTER TO FIND OUT THE JUNCTION WHICH HAS THE
C     LEAST NON-ZERO CIRCULATING TO NON CIRCULATING AIRFLOW RATIO.
C
            IF ((QREC*2.0/QIN).LE.NREC) THEN
               DO 30 L=1,M
                  MRC=MRC+1
                  K=MEMREC(L)
                  NOREC(MRC)=K
                  ESTPR(MRC)=AVRPR
                  RDPROP(K)=ESTPR(MRC)
                  ESTCH4(MRC)=AVRCH4
                  RDCH4(K)=ESTCH4(MRC)
                  ESTTR(MRC)=AVTRD
                  TRD(K)=ESTTR(MRC)
                  JF(K)=-JF(K)
30             CONTINUE
               L=I
               CALL CDJUNC (NSTART,JUMPBK,L,1)
               RETURN
            ENDIF
         ENDIF
60    CONTINUE
      IF (LJ.LT.NJ) THEN
         NREC=NREC+1
         GO TO 10
      ENDIF
```

```
C
      RETURN
      END
C
C
      SUBROUTINE  PREP (MRC,LST)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSE:
C        PREPARATION OF CALCULATED INTERMEDIATE RESULTS AS THE INITIAL
C     VALUES FOR THE NEXT ROUND OF ITERATIONS IN THE CASE WHEN
C     RECIRCULATION HAPPENS.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
      DO 10 I=1,NB
         JF(I)=IABS(JF(I))
10    CONTINUE
      DO 20 I=1,NJ
         JNO(I)=IABS(JNO(I))
20    CONTINUE
      IF (MRC.EQ.0) RETURN
      DO 30 I=1,MRC
         K=NOREC(I)
         DIFPR=ESTPR(I)-RDPROP(K)
         DIFCH4=ESTCH4(I)-RDCH4(K)
         DIFTRD=ESTTR(I)-TRD(K)
C
C        WHEN THE VARIATIONS IN TEMP. OR CONTAMINANT CONTENT ARE LARGER
C     THAN THE USER-SPECIFIED CRITERIA (LST>0), MORE ITERATIONS WILL BE
C     PERFORMED.
C
         IF (ABS(DIFPR).GE.(CRITSM/100.)) LST=LST+1
         IF (ABS(DIFCH4).GE.(CRITGS/100.)) LST=LST+1
         IF (ABS(DIFTRD).GE.(CRITHT)) LST=LST+1
C
C        CALCULATED AIR TEMP., CH4 AND FUME CONTENT AT AIRWAY ENDING
C     ARE USED AS NEW ESTIMATIONS FOR THE NEXT ITERATION.
C
         ESTPR(I)=RDPROP(K)
         ESTCH4(I)=RDCH4(K)
         ESTTR(I)=TRD(K)
         JF(K)=-JF(K)
30    CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE  TEVAL
C
```

```
C    ------------------------------------------------------------------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       1) EVALUATION OF MEAN AND MEAN-SQUARE AIR TEMP.
C       2) CALCULATION OF THE PRODUCT OF MEAN AIR TEMP. AND ELEVATION
C          VARIATION OF EACH AIRWAY.
C       3) EVALUATION OF VARIED AIRFLOW RESISTANCE DUE TO VARIED AIR
C          TEMP.
C       THIS SUBROUTINE IS CALLED FOR IN THE QUASI-EQUILIBRIUM
C    SIMULATION PART OF THE PROGRAM. ITS COUNTERPART IN THE DYNAMIC
C    SIMULATION PART IS PARAM.
C
C    ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
      DOUBLE PRECISION G
C
      CP=0.24
      DO 40 I=1,NB
         E=DZRD(I)/(2.*778.26*CP)
         B=TJS(I)-TROCK(I)
         G=XNEW(I)*LA(I)/CP
         IF (G.GE.25.0) THEN
            GX=0.0
            GXX=0.0
         ELSE
            GX=DEXP(-G)
            GXX=DEXP(-2.*G)
         ENDIF
         TMRD(I)=TROCK(I)-B*(GX-1.0)/G-E*(0.5-(GX+(GX-1.0)/G)/G)
         FRNVP(I)=(TMRD(I)+460.0)*DZRD(I)
         FFRNVP(I)=DZRD(I)/(TMRD(I)+460.0)
         TRA1=TROCK(I)+460.0
         A1=2*E**2/(2*G)**3+2*E*B/(2*G)**2-B**2/(2*G)
         A2=4*E**2/G**3+(920.*E+2*E*TJS(I))/G**2-2*TRA1*B/G
         A3=2*E**2/G+4*E**2/G**2+(920.*E+2*E*TJS(I))/G
         A4=E**2/(2*G)+2*E**2/(2*G)**2+E*B/G
         TMSQR=TRA1**2-TRA1*E-E*LA(I)/3.0+A1*(GXX-1.0)
     .                  +A2*(GX-1.0)+A3*GX+A4*GXX
         IF (NWTYP(I).NE.1) R(I)=RSTD(I)*TMSQR/(460.+TR)**2
40    CONTINUE
C
      RETURN
      END
C
C
     SUBROUTINE  NVP2
C
C    ------------------------------------------------------------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       EVALUATION OF NVP ACCORDING TO CALCULATED MEAN AIR TEMP. IN
C    AIRWAYS (REFER TO SUBS. TEVAL AND PARAM).
C
```

```
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE: 'CMMN.DAT'
C
      MBEGW=1
      SUMFNV=0.0
      DO 20 K=1,MNO
         MENDW=MEND(K)
C
C     FNTM: AVERAGE TEMPERATURE IN THE MESH.
C     FNVP(K): NVP IN MESH K. IN IN.W.G.
C     SUMFNV: SUM OF NVP VARIATION IN MESHES IN THE SUCCESSIVELY
C               PERFORMED ITERATIONS, SERVING AS A CRITERION TO JUDGE IF
C               THE CONVERGENT RESULTS ARE REACHED.
C
         ONVP=FNVP(K)
         FNVP(K)=0.
         ZUP=0.0
         ZDOWN=0.0
         HSU=0.
         TSU=0.
         DO 10 J=MBEGW,MENDW
            N=MSL(J)
            IF (N.LT.0) THEN
               FACT=-1.
               NX=-N
            ELSE
               FACT=1.
               NX=N
            ENDIF
            HSU=HSU+FACT*FRNVP(NX)
            TSU=TSU+FACT*FFRNVP(NX)
10       CONTINUE
         IF (ABS(TSU).GT.1.E-3) THEN
            HTU=ABS(HSU/TSU)
            FNTM=SQRT(HTU)
            FNVP(K)=HSU*DR/(5.2*FNTM)
         ELSE
            FNVP(K)=0.0
         ENDIF
         SUMFNV=SUMFNV+ABS(ONVP-FNVP(K))
         MBEGW=MENDW+1
20    CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE CDCH (NSTOP)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSE:
C     UPDATING CONDITION CHANGES ACCORDING TO USER-SPECIFIED TIME TABLE
C
```

```
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
      DIMENSION  QFX(20),PFX(20),DAL(15)
C
C
C
C       ICODE: CODE NUMBER. 1: CHANGE TO ORDINARY AIRWAY.   2: CHANGE
C              TO FAN BRANCH.  3: CHANGE TO FIRE SOURCE. 4: FIRE CEASED.
C              5: CHANGE SIMULATION STEPS. 6: CHANGE TIME INTERVAL FOR
C              OUTPUT. 7: JUNCTION WHOSE DOWN-STREAM SIDE AIRWAYS HAVE
C              DETAILED DATA RECORDS.
C       NBR: AIRWAY CALLING NUMBER IN WHICH CONDITION CHANGE HAPPENS.
C
4     CALL READIN (DAL,14,ISTOP,0)
      IF (ISTOP.EQ.1) THEN
         ICODE=DAL(2)
         WRITE (8,300)
         IF (ICODE.EQ.1) WRITE (8,310)
         IF (ICODE.EQ.2) WRITE (8,320)
         IF (ICODE.EQ.3) WRITE (8,330)
         IF (ICODE.EQ.4) WRITE (8,340)
         IF (ICODE.EQ.5) WRITE (8,350)
         IF (ICODE.EQ.6) WRITE (8,360)
         IF (ICODE.EQ.7) WRITE (8,365)
         WRITE (8,370) (ROW(IE),IE=1,80)
         NSTOP=5
         RETURN
      ENDIF
      IF (DAL(15).LT.(-1.E20)) GO TO 4
      IF (NCOMTS.GT.NCOMT2) THEN
         NCOMT2=NCOMTS
         GO TO 4
      ENDIF
      TS=DAL(1)
      IF (TS*60.0.GT.TACC) GO TO 100
      ICODE=DAL(2)
      NBR=DAL(3)
      DO 5 I=1,NB
          IF (NO(I).EQ.NBR) THEN
             N=I
             GO TO 6
          ENDIF
5     CONTINUE
      GO TO 4
C
C        ICODE=1 INDICATES THAT AIRWAY NBR BECOMES AN ORDINARY AIRWAY
C     WITH RESISTANCE EQUAL TO RCH AT TS MIN. AFTER EVENT.
C
6     IF (ICODE.EQ.1) THEN
         RSTD(N)=DAL(4)
         R(N)=RSTD(N)*(TMRD(N)+460.0)**2/(TR+460.0)**2
```

```
      NWTYP(N)=0
      IF (NFNUM.GT.0) THEN
        DO 8 J=1,NFNUM
          IF (NOF(J).EQ.NBR) THEN
            IF (NFNUM.GT.J) THEN
              DO 7 K=J,NFNUM-1
                NOF(K)=NOF(K+1)
                MPTS(K)=MPTS(K+1)
                NSKP(K)=NSKP(K+1)
7             CONTINUE
            ENDIF
            NOF(NFNUM)=0
            MPTS(NFNUM)=0
            NSKP(NFNUM)=0
            NFNUM=NFNUM-1
            GO TO 4
          ENDIF
8       CONTINUE
      ENDIF
      GO TO 4
C
C     ICODE=2 INDICATES THAT AIRWAY NBR BECOMES A FAN BRANCH WITH
C   CHARACTERISTICS.
C
    ELSE IF (ICODE.EQ.2) THEN
      INDEX=DAL(4)
      IF (INDEX.LE.0) THEN
        GO TO 4
      ELSE IF (INDEX.EQ.1) THEN
        RX=DAL(5)
        IF (NFNUM.GT.0) THEN
          DO 50 J=1,NFNUM
            IF (NOF(J).EQ.NBR) THEN
              DO 40 JJ=J,NFNUM-1
                NOF(JJ)=NOF(JJ+1)
                MPTS(JJ)=MPTS(JJ+1)
                DO 40 KK=1,IMX
                  QF(JJ,KK)=QF(JJ+1,KK)
                  PF(JJ,KK)=PF(JJ+1,KK)
40            CONTINUE
              NFNUM=NFNUM-1
            ENDIF
50        CONTINUE
        ENDIF
        NWTYP(N)=1
        R(N)=RX
        GO TO 4
      ELSE IF (INDEX.GT.5) THEN
        INDEX=5
      ENDIF
      DO 55 I=1,INDEX
        II=5+(I-1)*2
        QFX(I)=DAL(II)
        PFX(I)=DAL(II+1)
```

```
55      CONTINUE
        IF (NFNUM.GT.0) THEN
          DO 20 J=1,NFNUM
             IF (NOF(J).EQ.NO(N)) THEN
                MPTS(J)=INDEX
                DO 10 K=1,INDEX
                   QF(J,K)=QFX(K)
                   PF(J,K)=PFX(K)
10              CONTINUE
                JN=J
                GO TO 4
             ENDIF
20        CONTINUE
        ENDIF
        NFNUM=NFNUM+1
        IF (NFNUM.GT.IMX) THEN
          NFNUM=NFNUM-1
          NWTYP(N)=1
          R(N)=PFX(1)
          GO TO 4
        ENDIF
        NWTYP(N)=1
        NOF(NFNUM)=NO(N)
        MPTS(NFNUM)=INDEX
        DO 30 K=1,INDEX
           QF(NFNUM,K)=QFX(K)
           PF(NFNUM,K)=PFX(K)
30      CONTINUE
        GO TO 4
C
C       ICODE=3 INDICATES THAT BRANCH NBR BECOMES A FIRE SOURCE AT
C     TS MIN. AFTER EVENT.
C
      ELSE IF (ICODE.EQ.3) THEN
        CONTX=DAL(4)
        CONCX=DAL(5)
        HEATX=DAL(6)
        O2MINX=DAL(7)
        SMPO2X=DAL(8)
        HTPO2X=DAL(9)
        QCENTX=DAL(10)
        TPRX=DAL(11)
        IF (INFLOW.GT.0) THEN
          DO 60 J=1,INFLOW
             IF (NCENT(J).EQ.NBR) THEN
                CONT(J)=CONTX
                CONC(J)=CONCX
                HEAT(J)=HEATX
                O2MIN(J)=O2MINX
                SMPO2(J)=SMPO2X
                HTPO2(J)=HTPO2X
                QCENT(J)=QCENTX
                TPR(J)=TPRX
                JN=J
```

```
                GO TO 4
              ENDIF
60          CONTINUE
          ENDIF
          INFLOW=INFLOW+1
          IF (INFLOW.GT.IMX) GO TO 4
          NCENT(INFLOW)=NBR
          CONT(INFLOW)=CONTX
          CONC(INFLOW)=CONCX
          HEAT(INFLOW)=HEATX
          O2MIN(INFLOW)=O2MINX
          SMPO2(INFLOW)=SMPO2X
          HTPO2(INFLOW)=HTPO2X
          QCENT(INFLOW)=QCENTX
          TPR(INFLOW)=TPRX
          JN=INFLOW
          GO TO 4
C
C       ICODE=4 INDICATES THAT BRANCH NBR BECOMES AN ORDINARY AIRWAY
C     WITHOUT A FIRE SOURCE AT TS MIN. AFTER EVENT.
C
      ELSE IF (ICODE.EQ.4) THEN
        IF (INFLOW.GT.0) THEN
          DO 90 I=1,INFLOW
              IF (NCENT(I).EQ.NBR) THEN
                DO 80 J=I,INFLOW
                    NCENT(J)=NCENT(J+1)
                    CONT(J)=CONT(J+1)
                    CONC(J)=CONC(J+1)
                    HEAT(J)=HEAT(J+1)
                    O2MIN(J)=O2MIN(J+1)
                    SMPO2(J)=SMPO2(J+1)
                    HTPO2(J)=HTPO2(J+1)
                    QCENT(J)=QCENT(J+1)
                    TPR(J)=TPR(J+1)
80              CONTINUE
                INFLOW=INFLOW-1
                NWTYP(N)=0
                GO TO 4
              ENDIF
90        CONTINUE
        ENDIF
      ENDIF
      GO TO 4
100   RETURN
C
C
300   FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ',
     .        ' IN:',//)
310   FORMAT (1X,'TS,1,NBR,R')
320   FORMAT (1X,'TS,2,NBR,INDEX,QF,PF,QF,PF,QF,PF,QF,PF,QF,PF')
330   FORMAT (1X,'TS,3,NBR,CONT,CONC,HEAT,02MIN,SMPO2,HTPO2,QCENT')
340   FORMAT (1X,'TS,4,NBR')
350   FORMAT (1X,'TS,5,DELTAT')
```

```
360  FORMAT (1X,'TS,6,TOUT')
365  FORMAT (1X,'TS,7,JDPP')
370  FORMAT (/,1X,'THE CURRENT INPUT WAS:',//,80A1)
C
     END
C
C

C
     SUBROUTINE READIN (DAL,MAX,ISTOP,ICT)
C
C    ------------------------------------------------------------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       DATA INPUT.
C
C    ------------------------------------------------------------------------------------------------------------------------
C
     COMMON /CHRC/ ITITLE(50),NCOMT1,NCOMTS,NCOMT2,LLINE
     COMMON /ERMSG/ ROW(80),TITLE(50,20)
     CHARACTER*4 TITLE
     CHARACTER*1 ROW,CHK(15),ER(80)
     DIMENSION DAL(15)
     DATA CHK/'0','1','2','3','4','5','6','7','8','9','.',',','+',
    .               '-',' '/
C
C
     KK=0
     KK1=0
     KK2=0
     NEG=0
     ISTOP=0
     DO 5 I=1,15
        DAL(I)=0.0
5    CONTINUE
     DO 7 I=1,80
        ER(I)=' '
7    CONTINUE
     READ (9,110) (ROW(I),I=1,80)
     LLINE=LLINE+1
     IDEC=0
     MLEV=1
     JJ=0
     I=15
     LEV=0
10   LEV=LEV+1
     II=I
     IF (LEV.GT.80) GO TO 50
     DO 20 I=1,15
        IF (ROW(LEV).EQ.CHK(I)) THEN
           IF (I.NE.15) KK=1
           GO TO 30
        ENDIF
20   CONTINUE
```

```
      KK2=KK2+1
      IF ((KK.EQ.0.AND.ROW(LEV).NE.'$').OR.KK2.GE.4) THEN
         ISTOP=0
         NCOMTS=NCOMTS+1
         IF (NCOMTS.LE.50) THEN
            IF (ICT.EQ.1) THEN
               KK3=0
               KK4=0
               DO 21 I=1,80
                  IF (ROW(I).NE.CHK(15)) GO TO 22
                  KK3=KK3+1
21             CONTINUE
22             DO 23 I=80,1,-1
                  IF (ROW(I).NE.CHK(15)) GO TO 24
                  KK4=KK4+1
23             CONTINUE
24             LR=(KK3+KK4-8)/2
               IF (LR.LT.0) LR=0
               KK5=80-KK4
               IF (LR.EQ.0) KK5=KK3+72
               DO 25 I=KK3+1,KK5
                  ER(I-KK3+LR)=ROW(I)
25             CONTINUE
               DO 26 I=1,80
                  IF (I.LE.LR) THEN
                     ROW(I)=' '
                  ELSE
                     ROW(I)=ER(I)
                  ENDIF
26             CONTINUE
            ENDIF
            DO 27 I=1,20
               J=(I-1)*4+1
               TITLE(NCOMTS,I)=ROW(J)//ROW(J+1)//ROW(J+2)//ROW(J+3)
27          CONTINUE
         ENDIF
         RETURN
      ELSE IF (KK.EQ.0.AND.ROW(LEV).EQ.'$') THEN
         KK1=1
         GO TO 50
      ELSE IF (KK1.NE.1) THEN
         ER(LEV)=ROW(LEV)
         ISTOP=1
         I=1
      ENDIF
30    IF (I.GT.10) GO TO 40
      IF (JJ.EQ.1) MLEV=MLEV+1
      JJ=0
      IF (IDEC.EQ.0) THEN
         DAL(MLEV)=(DAL(MLEV)*10.0)+((I-1)*1.0)
      ELSE IF (IDEC.GT.0) THEN
         DAL(MLEV)=DAL(MLEV)+((I-1)*1.0)/(10.0**IDEC)
      ENDIF
      IF (IDEC.NE.0) IDEC=IDEC+1
```

```
      GO TO 10
40    IF (I.EQ.11) IDEC=1
      IF (I.EQ.14) NEG=1
      IF (I.EQ.12.OR.I.EQ.15) THEN
         IF (NEG.EQ.1) DAL(MLEV)=0.0-DAL(MLEV)
         NEG=0
         IDEC=0
      ENDIF
      IF (JJ.EQ.1.AND.(I.EQ.11.OR.I.EQ.13.OR.I.EQ.14)) THEN
         MLEV=MLEV+1
         JJ=0
      ENDIF
      IF (I.EQ.12) THEN
         MLEV=MLEV+1
         JJ=0
      ENDIF
      IF (I.EQ.15.AND.II.EQ.12) JJ=5
      IF (I.EQ.15.AND.II.NE.15.AND.II.NE.12.AND.JJ.EQ.0) JJ=1
      IF (I.EQ.12) DAL(MLEV)=0.0
      IF (MLEV.LE.MAX) GO TO 10
50    IF (ISTOP.GT.0) WRITE (8,100) (ROW(I),I=1,80),(ER(I),I=1,80)
      IF (KK.EQ.0.OR.KK1.EQ.1) DAL(15)=-1.E25
      RETURN
C
100   FORMAT (/,' * ERROR *  ILLEGAL CHARACTER DETECTED IN ',
     .        'THE LINE WHERE NUMERICAL INPUT',/,' WAS EXPECTED. ',
     .        'TWO LINES FOLLOW: 1 ORIGINAL; 2 ILLEGAL CHARACTERS.',
     .        //,80A1,/,80A1,/)
110   FORMAT (80A1)
C
      END
C
      SUBROUTINE OUTPUT (IPT,MARKX,MARKY,MADJC)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSE:
C        OUTPUT OF NORMAL CALCULATION RESULTS.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
      DIMENSION HFN(5),QFN(5),LST(NMX),TQR(NMX)
C
C
      IF (IPT.NE.1) THEN
         PP=0.0
         DO 10 L=1,NB
            IF (NWTYP(L).LE.0.OR.NWTYP(L).EQ.10) THEN
               P(L)=R(L)*Q(L)*ABS(Q(L))*1.E-10
            ELSE
               P(L)=R(L)
            ENDIF
```

```
          PP=PP+P(L)
10      CONTINUE
        PCRIT=PP/FLOAT(NB)
C
        IF (MADJC.GE.MADJ.AND.IPT.EQ.4) THEN
          DO 1520 I=1,NB
            TTMN=0.0
            DO 1500 J=1,10
              TTMN=TTMN+TMRCD(I,J)
1500        CONTINUE
            TTMN=TTMN/10.0
            TQR(I)=0.0
            DO 1510 J=1,10
              TQR(I)=TQR(I)+(TMRCD(I,J)-TTMN)**2
1510        CONTINUE
1520      CONTINUE
          IZ=0
          LL=0
1530      II=0
          DO 1540 I=1,NB
            IF (NO(I).GT.0) THEN
              IF (TQR(I).GT.TSR.OR.II.EQ.0) THEN
                TSR=TQR(I)
                II=I
              ENDIF
            ENDIF
1540      CONTINUE
          IF (II.EQ.0) GO TO 1590
          IF (LL.GT.0) THEN
            KK=0
            DO 1550 J=1,LL
              K=LST(J)
              IF (TQR(II).GT.TQR(K)) KK=KK+1
1550        CONTINUE
            IF (KK.GE.(LL-2).AND.TQR(II).GT.20.0) GO TO 1560
            GO TO 1590
          ENDIF
1560      LL=LL+1
          LST(LL)=II
          NO(II)=-NO(II)
          JSS=JS(II)
          JFF=JF(II)
1570      DO 1580 I=1,NB
            IF (NO(I).GT.0) THEN
              IF ((JS(I).EQ.JSS.OR.JF(I).EQ.JSS.OR.JS(I).EQ.JFF.OR.
     .          JF(I).EQ.JFF).AND.(TQR(I).GT.20.0)) THEN
                LL=LL+1
                LST(LL)=I
                NO(I)=-NO(I)
                IF (JS(I).EQ.JSS) THEN
                  JSS=JF(I)
                ELSE IF (JF(I).EQ.JSS) THEN
                  JSS=JS(I)
                ELSE IF (JS(I).EQ.JFF) THEN
```

```
                    JFF=JF(I)
                  ELSE
                    JFF=JS(I)
                  ENDIF
                  GO TO 1570
                ENDIF
              ENDIF
1580        CONTINUE
            IZ=IZ+1
            IF (IZ.LE.3) GO TO 1530
1590        CRT1=0.0
            CRT2=0.0
            CRT3=0.0
            DO 1630 I=1,LL
              K=LST(I)
              NO(K)=IABS(NO(K))
              CRT=0.0
              DO 1600 J=10,3,-1
                CRT=CRT+(TMRCD(K,J)-TMRCD(K,J-2))**2
1600          CONTINUE
              CRT1=CRT1+CRT/8.0
              CRT=0.0
              DO 1610 J=10,4,-1
                CRT=CRT+(TMRCD(K,J)-TMRCD(K,J-3))**2
1610          CONTINUE
              CRT2=CRT2+CRT/7.0
              CRT=0.0
              DO 1620 J=10,5,-1
                CRT=CRT+(TMRCD(K,J)-TMRCD(K,J)-TMRCD(K,J-4))**2
1620          CONTINUE
              CRT3=CRT3+CRT/6.0
1630        CONTINUE
            CRT1=CRT1/LL
            CRT2=CRT2/LL
            CRT3=CRT3/LL
            LSG=LL
            IF (MIN(CRT1,CRT2,CRT3)*10.0.LE.MAX(CRT1,CRT2,CRT3)) THEN
              ICYCLE=2
              IF (CRT2.LT.0.95*CRT1) ICYCLE=3
              IF (CRT3.LT.0.9*CRT1.AND.CRT3.LT.CRT2) ICYCLE=4
            ELSE
              ICYCLE=0
            ENDIF
          ENDIF
        ENDIF
C
        IF (IPT.EQ.1) THEN
C
C
          WRITE (8,490)
          LINE=NCOMT1+1
          DO 15 K=1,NB
13          IJ=0
            IF (LINE.LE.50) IJ=ITITLE(LINE)
```

```
            IF (K.EQ.IJ) THEN
               IF (LINE.EQ.1) I1=0
               IF (LINE.NE.1) I1=ITITLE(LINE-1)
               IF (LINE.EQ.50) I2=0
               IF (LINE.NE.50) I2=ITITLE(LINE+1)
               IF (I2.NE.ITITLE(LINE).AND.I1.NE.ITITLE(LINE)) THEN
                  WRITE (8,930) (TITLE(LINE,J),J=1,18)
               ELSE IF (I2.EQ.ITITLE(LINE).AND.I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,960) (TITLE(LINE,J),J=1,18)
               ELSE IF (I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,940) (TITLE(LINE,J),J=1,18)
               ELSE
                  WRITE (8,950) (TITLE(LINE,J),J=1,18)
               ENDIF
               LINE=LINE+1
               GO TO 13
            ENDIF
            WRITE (8,500) NO(K),JS(K),JF(K),Q(K),NWTYP(K),LA(K),A(K),
                          R(K),KF(K),O(K)
15       CONTINUE
         IF (NVPN.LE.0) THEN
            WRITE (8,980)
            WRITE (8,990) (JNO(K),T(K),Z(K),CH4C(K),K=1,NJ)
         ENDIF
C
      ELSE IF (IPT.EQ.2) THEN
C
C
         WRITE (8,110)
         WRITE (8,120)
         L=0
         LINE=NCOMT1+1
         DO 20 K=1,NB
18          IJ=0
            IF (LINE.LE.50) IJ=ITITLE(LINE)
            IF (K.EQ.IJ) THEN
               IF (LINE.EQ.1) I1=0
               IF (LINE.NE.1) I1=ITITLE(LINE-1)
               IF (LINE.EQ.50) I2=0
               IF (LINE.NE.50) I2=ITITLE(LINE+1)
               IF (I2.NE.ITITLE(LINE).AND.I1.NE.ITITLE(LINE)) THEN
                  WRITE (8,930) (TITLE(LINE,J),J=1,18)
               ELSE IF (I2.EQ.ITITLE(LINE).AND.I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,960) (TITLE(LINE,J),J=1,18)
               ELSE IF (I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,940) (TITLE(LINE,J),J=1,18)
               ELSE
                  WRITE (8,950) (TITLE(LINE,J),J=1,18)
               ENDIF
               LINE=LINE+1
               GO TO 18
            ENDIF
            IF (NWTYP(K).GT.0) THEN
               L=L+1
```

```
          ELSE
            WRITE (8,130) NO(K),JS(K),JF(K),NWTYP(K),R(K),Q(K),P(K)
          ENDIF
20      CONTINUE
        IF (L.LE.0) THEN
          WRITE (8,140)
        ELSE
          WRITE (8,150)
          DO 30 K=1,NB
              IF (NWTYP(K).LE.0) GO TO 30
              WRITE (8,160) NO(K),JS(K),JF(K),Q(K),P(K)
30        CONTINUE
          IF (NFNUM.GT.0) THEN
            WRITE (8,170) (NOF(K),K=1,NFNUM)
            WRITE (8,175)
            DO 40 K=1,NFNUM
                L=MPTS(K)
                WRITE (8,180) (QF(K,I)/1000.0,PF(K,I),I=1,L)
40          CONTINUE
            IF (IOUT.LE.(-2).AND.MARKY.LE.0) THEN
              WRITE (8,1040)
              DO 45 K=1,NFNUM
                  WRITE (8,1045) NOF(K)
                  INDEX=MPTS(K)
                  IF (INDEX.GT.1) THEN
                    DO 43 K1=1,INDEX-1
                        WRITE (8,1050) (FKQ(K,K1,K2),K2=1,4),QF(K,K1)
                                       ,QF(K,K1+1)
43                  CONTINUE
                    WRITE (8,1085)
                    DO 441 K1=1,INDEX-1
                        QZZ=(QF(K,K1+1)-QF(K,K1))/5.0
                        DO 44 K2=1,5
                            QZZ1=QF(K,K1)+(K2-1)*QZZ
                            NFCWZ=NFCW(K)
                            CALL SPLINE (K,QZZ1,HZZ,TR,1)
                            NFCW(K)=NFCWZ
                            HFN(K2)=HZZ
                            QFN(K2)=QZZ1
44                      CONTINUE
                        WRITE (8,1090) (QFN(K4)/1000.,HFN(K4),K4=1,5)
441                 CONTINUE
                  ENDIF
45            CONTINUE
            ENDIF
          ENDIF
        ENDIF
        WRITE (8,190) NB,NJ
        IF (NFNUM.GT.0) THEN
          DO 62 K=1,NFNUM
              DO 50 KK=1,NB
                  IF (NO(KK).EQ.NOF(K)) GO TO 60
50            CONTINUE
              GO TO 62
```

```
60            IF (NFCW(K).GT.0) WRITE (8,200) NOF(K)
              IF ((Q(KK).LT.0.0.AND.IABS(JS(KK)).EQ.JSB(KK)).OR.
     .           (Q(KK).GT.0.0.AND.IABS(JS(KK)).NE.JSB(KK)))
     .           WRITE (8,840) NOF(K)
62        CONTINUE
        ENDIF
      ELSE IF (IPT.EQ.3) THEN
C
C
        IF (IOUT.GT.0.OR.(IOUT.EQ.(-1).AND.MARKY.GE.1)) THEN
          WRITE (8,1060)
          LINE=NCOMT1+1
          DO 66 K=1,NB
64            IJ=0
              IF (LINE.LE.50) IJ=ITITLE(LINE)
              IF (K.EQ.IJ) THEN
                IF (LINE.EQ.1) I1=0
                IF (LINE.NE.1) I1=ITITLE(LINE-1)
                IF (LINE.EQ.50) I2=0
                IF (LINE.NE.50) I2=ITITLE(LINE+1)
                IF (I2.NE.ITITLE(LINE).AND.I1.NE.ITITLE(LINE)) THEN
                  WRITE (8,930) (TITLE(LINE,J),J=1,18)
                ELSE IF (I2.EQ.ITITLE(LINE).AND.I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,960) (TITLE(LINE,J),J=1,18)
                ELSE IF (I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,940) (TITLE(LINE,J),J=1,18)
                ELSE
                  WRITE (8,950) (TITLE(LINE,J),J=1,18)
                ENDIF
                LINE=LINE+1
                GO TO 64
              ENDIF
              WRITE (8,1070) NO(K),JS(K),JF(K),NWTYP(K),TROCK(K),LA(K),
     .                       DZRD(K),RSTD(K),HK(K),HA(K),CH4V(K)
66        CONTINUE
          IF (MARKY.LE.0) WRITE (8,1080)
        ENDIF
        IF (IOUT.LE.(-2).OR.(IOUT.EQ.(-1).AND.MARKY.LE.0)) THEN
          WRITE (8,800)
          LINE=NCOMT1+1
          DO 74 I=1,NB
72            IJ=0
              IF (LINE.LE.50) IJ=ITITLE(LINE)
              IF (I.EQ.IJ) THEN
                IF (LINE.EQ.1) I1=0
                IF (LINE.NE.1) I1=ITITLE(LINE-1)
                IF (LINE.EQ.50) I2=0
                IF (LINE.NE.50) I2=ITITLE(LINE+1)
                IF (I2.NE.ITITLE(LINE).AND.I1.NE.ITITLE(LINE)) THEN
                  WRITE (8,930) (TITLE(LINE,J),J=1,18)
                ELSE IF (I2.EQ.ITITLE(LINE).AND.I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,960) (TITLE(LINE,J),J=1,18)
                ELSE IF (I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,940) (TITLE(LINE,J),J=1,18)
```

```
            ELSE
               WRITE (8,950) (TITLE(LINE,J),J=1,18)
            ENDIF
            LINE=LINE+1
            GO TO 72
          ENDIF
          WRITE (8,810) NO(I),JS(I),JF(I),DZRD(I),TROCK(I),
     .                  CH4V(I),HK(I),HA(I)
74      CONTINUE
      ENDIF
      IF (MARKY.LE.0.OR.IOUT.NE.0) THEN
        WRITE (8,240) TSTART,JSTART
        IF (INFLOW.GT.0) THEN
          WRITE (8,250)
          WRITE (8,260) (NCENT(I),CONT(I),CONC(I),HEAT(I),O2MIN(I),
     .                   SMPO2(I),HTPO2(I),TPR(I),INT(QCENT(I)),I=1,INFLOW)
        ELSE
          WRITE (8,270)
        ENDIF
      ENDIF
    ELSE IF (IPT.EQ.4) THEN
C
C
      ITL=MADJC
      JTL=MADJ
      IF (MARKX.GE.1.AND.MARKY.LE.0) THEN
        WRITE (8,320) TACC
      ELSE IF (MARKY.GE.1) THEN
        WRITE (8,970) TIME
      ENDIF
      IF (ITL.GE.JTL) THEN
        WRITE (8,690) SUMFNV/MNO
        KTL=0
        DO 78 I=1,NB
          NREVA=NREV(I)
          NREVA=NNREV(I)
          IF (NREVA.GE.MADJ/3) THEN
            IF (KTL.EQ.0) THEN
              WRITE (8,700)
              KTL=1
            ENDIF
            IF (NWTYP(I).EQ.1) THEN
              PP=R(I)
            ELSE
              PP=R(I)*Q(I)*ABS(Q(I))*1.E-10
            ENDIF
            WRITE (8,710) NO(I),Q(I),PP,NREVA,ITL
          ENDIF
78      CONTINUE
        IZ=MADJ-9
        IF (ICYCLE.EQ.0) THEN
          WRITE (8,890) (IJ,IJ=IZ,IZ+9)
        ELSE
          WRITE (8,900) ICYCLE,(IJ,IJ=IZ,IZ+9)
```

```
80
          ENDIF
          WRITE (8,905) (HRM(IJ),IJ=1,10)
          DO 80 I=1,LSG
             K=LST(I)
             IF (JSB(K).EQ.JS(K)) THEN
                JSS=JS(K)
                JFF=JF(K)
             ELSE
                JSS=JF(K)
                JFF=JS(K)
             ENDIF
             WRITE (8,910) NO(K),JSS,JFF,(TMRCD(K,J),J=1,10)
             WRITE (8,920) (QRCD(K,J)/1000.0,J=1,10)
80        CONTINUE
       ELSE
          IF (SUMFNV/MNO.LE.2.E-4) WRITE (8,770) SUMFNV/MNO,ITL
          IF (SUMFNV/MNO.GT.2.E-4) WRITE (8,780) SUMFNV/MNO,ITL
       ENDIF
       LINE=NCOMT1+1
       WRITE (8,300)
       DO 84 I=1,NB
82        IJ=0
          IF (LINE.LE.50) IJ=ITITLE(LINE)
          IF (I.EQ.IJ) THEN
             IF (LINE.EQ.1) I1=0
             IF (LINE.NE.1) I1=ITITLE(LINE-1)
             IF (LINE.EQ.50) I2=0
             IF (LINE.NE.50) I2=ITITLE(LINE+1)
             IF (I2.NE.ITITLE(LINE).AND.I1.NE.ITITLE(LINE)) THEN
                WRITE (8,930) (TITLE(LINE,J),J=1,18)
             ELSE IF (I2.EQ.ITITLE(LINE).AND.I1.EQ.ITITLE(LINE)) THEN
                WRITE (8,960) (TITLE(LINE,J),J=1,18)
             ELSE IF (I1.EQ.ITITLE(LINE)) THEN
                WRITE (8,940) (TITLE(LINE,J),J=1,18)
             ELSE
                WRITE (8,950) (TITLE(LINE,J),J=1,18)
             ENDIF
             LINE=LINE+1
             GO TO 82
          ENDIF
          WRITE (8,310) NO(I),IABS(JS(I)),IABS(JF(I)),Q(I),TMRD(I),
     .                  TRD(I),RDPROP(I),RDCH4(I),P(I)
84     CONTINUE
       IF (INFLOW.GT.0) THEN
          WRITE (8,820)
          DO 89 L=1,INFLOW
             DO 88 J=1,NB
                IF (NCENT(L).EQ.NO(J)) THEN
                   WRITE (8,830) NO(J),Q(J),TJS(J),RDPROP(J),
     .                           O2BEH(L),HTAD(L)
                   IF (TJS(J).GT.3500.0) WRITE (8,995) NO(J)
                   GO TO 89
                ENDIF
88           CONTINUE
```

```
89          CONTINUE
          ENDIF
          IF (IOUT.NE.0) THEN
            WRITE (8,650)
            WRITE (8,660) (JNO(I),T(I),PROP(I),PRCH4(I),I=1,NJ)
          ENDIF
          KTL=0
          DO 90 I=1,NB
             NNREVA=NNREV(I)
             IF ((NNREVA/2)*2.NE.NNREVA) THEN
               IF (KTL.EQ.0) THEN
                 WRITE (8,470)
                 KTL=1
               ENDIF
               WRITE (8,480) NO(I),IABS(JS(I)),IABS(JF(I))
             ENDIF
90        CONTINUE
       ELSE IF (IPT.EQ.5) THEN
          WRNSUM=WRNPR+WRNGS+WRNSM+WRNHT
          IF (WRNSUM.LE.0.) THEN
            WRITE (8,410)
          ELSE
            J=0
            WRNP=WRNPR
            IF (PCRIT*0.05.LE.0.5*WRNPR) WRNP=PCRIT*0.05
            DO 95 I=1,NB
               K=0
               L=0
               M=0
               N=0
               IF ((100.*RDCH4(I)).GE.WRNGS) K=K+1
               IF ((100.*RDPROP(I)).GE.WRNSM) L=L+1
               IF (TRD(I).GE.WRNHT) M=M+1
               IF (P(I).LT.WRNP) N=N+1
               IF ((K+L+M+N).GT.0) THEN
                 IF (J.EQ.0) THEN
                   WRITE (8,420) WRNGS,WRNSM,WRNHT,WRNP
                   J=J+1
                 ENDIF
                 WRITE (8,430) NO(I),IABS(JS(I)),IABS(JF(I)),RDCH4(I),
                               RDPROP(I),TRD(I),P(I)
               ENDIF
95          CONTINUE
            IF (J.LE.0) WRITE (8,440)
            J=0
            KM=0
            DO 100 I=1,NJ
               K=0
               L=0
               M=0
               IF ((100.*PRCH4(I)).GE.WRNGS) K=K+1
               IF ((100.*PROP(I)).GE.WRNSM) L=L+1
               IF (T(I).GE.WRNHT) M=M+1
               IF ((K+L+M).GT.0) THEN
```

```
                IF (J.EQ.0) THEN
                   WRITE (8,450) WRNGS,WRNSM,WRNHT,WRNGS,WRNSM,WRNHT
                ENDIF
                J=J+1
                IF (2*(J/2).NE.J) THEN
                   KM=I
                ELSE
                   WRITE (8,460) JNO(KM),PRCH4(KM),PROP(KM),T(KM),
     .                           JNO(I),PRCH4(I),PROP(I),T(I)
                   KM=0
                ENDIF
             ENDIF
100       CONTINUE
          IF (KM.GT.0) WRITE (8,460) JNO(KM),PRCH4(KM),PROP(KM),T(KM)
          IF (J.LE.0) WRITE (8,445)
       ENDIF
    ENDIF
C
    RETURN
C
110  FORMAT (///,22X,'NETWORK CALCULATION RESULTS ',/)
120  FORMAT (1X,'AIRWAY',2X,'FROM',4X,'TO',3X,'TYPE',8X,'RESISTANCE',
    .        7X,'AIRFLOW',6X,'HEADLOSS',/)
130  FORMAT (I5,I7,I6,I7,8X,E9.4,6X,F10.0,3X,F10.3)
140  FORMAT (///,1X,'NETWORK CONTAINS NO FANS')
150  FORMAT (///,T20,'FANS'//' AIRWAY  FROM    TO     AIRFLOW    FAN',
    .        ' PRESSURE',/)
160  FORMAT (I5,I7,I7,F13.0,F13.3)
170  FORMAT (/////,' THESE CHARACTERISTICS WERE STORED FOR FANS',5I5,/
    .         T45,5I5)
175  FORMAT (//,5('  Q*1000   PF '),/)
180  FORMAT (5(F8.1,F6.2)/,5(F8.1,F6.2),/)
190  FORMAT (///,1X,'THE STATED NUMBER OF AIRWAYS WAS',I9,/,1X,'THE ',
    .        'STATED NUMBER OF JUNCTIONS WAS',I7)
200  FORMAT (///,1X,'FAN CHARACTERISTICS ARE EXCEEDED FOR FAN NO.',I5)
240  FORMAT (///,1X,'A TEMPERATURE OF',F6.1,' WAS ASSIGNED TO ',
    .        'JUNCTION NO.',I7)
250  FORMAT (////,T18,'THE FOLLOWING CONTAMINATION WAS ASSUMED',//,
    .        12X,'FIXED INFLUX',T31,'O2-RICH',T41,'FUEL-RICH',T55,
    .        'LEADING T',T67,'REF. Q',/,' AIRWAY  FLUX  %CONC',
    .        3X,'HEAT',5X,'%O2',4X,'FUMES',2X,'HEAT',/)
260  FORMAT (I5,F8.1,F6.2,1X,E9.3,F7.2,F7.2,F7.1,F10.2,I12)
270  FORMAT (///,1X,' NO CONTAMINATION WAS SPECIFIED')
300  FORMAT (///,T8,'TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN'
    .        ,' AIRWAYS',//,' AIRWAY',T9,'FROM',T15,'TO',T20,'AIRFLOW',
    .        T31,'AVE. T',T40,'T AT END',T50,'FUMES',T57,'METHANE',T65,
    .        'HEADLOSS',/)
310  FORMAT (I5,I6,I5,T18,0PF9.0,T30,0PF7.2,T40,0PF7.2,T49,2PF6.2,
    .        T56,2PF6.2,T66,0PF6.3)
320  FORMAT (///,20X,'TIME AT ',F6.0,' SEC. AFTER EVENT',//)
410  FORMAT (///,1X,'NO THRESHOLD FOR CRITICAL STATES ',
    .        'WERE SPECIFIED.')
420  FORMAT (///,9X,'IN THE FOLLOWING AIRWAYS EXIST CRITICAL ',
    .        'CONDITIONS',//,1X,'AIRWAY',T10,'FROM',T16,'TO',T22,'CH4 %'
```

```
     .          ,T33,'FUMES %',T45,'TEMPERATURE',T61,'HEADLOSS',/,
     .          T22,'>',F5.3,T33,'>',F6.3,T46,'>',F5.0,' F',T59,
     .          '<',F6.3,' IN.WG.',/)
430  FORMAT (I5,I7,I5,T21,2PF6.2,T32,2PF8.3,T45,0PF8.1,T62,0PF6.3)
440  FORMAT (///,1X,'NO CRITICAL CONDITIONS AT AIRWAY ENDS ',
     .          'WERE DETECTED')
445  FORMAT (///,1X,'NO CRITICAL CONDITIONS IN JUNCTIONS WERE ',
     .          'DETECTED.')
450  FORMAT (///,10X,'IN THE FOLLOWING JUNCTIONS EXIST CRITICAL ',
     .          'CONDITIONS',//,' JUNCTION',T11,'CH4  %',T19,'FUMES %',T28
     .          ,'TEMP. F',T40,'JUNCTION',T49,'CH4  %',T57,'FUMES %',T66,
     .          'TEMP. F',/,T11,'>',F5.2,T19,'>',F6.3,T29,'>',F5.0,T49,
     .          '>',F5.2,T57,'>',F6.3,T67,'>',F5.0,/)
460  FORMAT (I6,T11,2PF5.2,T17,2PF8.2,T27,0PF7.1,T39,I6,T49,2PF5.2,
     .          T55,2PF8.2,T65,0PF7.1)
470  FORMAT (///,1X,'REVERSAL OF AIRFLOW HAS OCCURRED IN THE ',
     .          'FOLLOWING PLACES',/)
480  FORMAT (1X,'AIRWAY',I6,' IS NOW CARRYING AIR FROM',I6,' TO',I6)
490  FORMAT (/////,3X,'NO NETWORK CALCULATION HAS BEEN PERFORMED, ',
     .          'THESE ARE THE INPUT DATA',//,' AIRWAY FROM',T15,'TO',
     .          T22,'AIRFLOW',T30,'TYPE',T37,'LENGTH',T47,'AREA',
     .          T57,'R',T64,'K',T68,'PERI.',/)
500  FORMAT (I5,I6,T13,I4,T19,F10.0,T30,I3,T36,F7.1,T44,F7.1,T53,E8.3,
     .          T62,I4,F7.1)
650  FORMAT (///,19X,'PARAMETERS OF AIR IN JUNCTIONS',//,1X,'JUNCTION'
     .          ,T13,'TEMP.',T20,'FUMES',T27,'METHANE',T42,'JUNCTION',T53,
     .          'TEMP.',T60,'FUMES',T67,'METHANE',/)
660  FORMAT (I6,T9,0PF9.2,T18,2PF7.2,T26,2PF7.2,T41,I6,T49,0PF9.2,T58,
     .          2PF7.2,T66,2PF7.2)
690  FORMAT (///,1X,'THE CALCULATION WAS NOT COMPLETED SINCE THE ',
     .          'NUMBER OF EXCHANGES BETWEEN',/,' NETWORK AND TEMPERATURE'
     .          ,' PARTS OF THE PROGRAM BECAME EXCESSIVE.',/,' THRESHOLD',
     .          ' IN NVP IS 0.0002 IN.W.G. PER MESH. THE NVP CORRECTION ',
     .          'PER'/,' MESH NOW EQUALS ',F7.5,' IN.W.G.')
700  FORMAT (///,1X,'THE FOLLOWING UNSTABLE AIRWAYS MAY HAVE',1X,
     .          'CAUSED THE INCOMPLETION',/,' OF NETWORK BALANCING:',/)
710  FORMAT (1X,'AIRWAY',I5,' OF AIRFLOW RATE',F7.0,' (CFM) AND ',
     .          'HEADLOSS',F7.3,' IN.W.G.',/,' HAD ITS AIRFLOW DIRECTION'
     .          ,' REVERSED',I3,' TIMES IN',I3,' ITERATIONS')
770  FORMAT (/////,' THRESHOLD IN ACCURACY (SUM OF NVP CORRECTIONS ',
     .          'PER MESH < 2.E-4 IN.W.G.)',/,' SATISFIED. CURRENT ',
     .          'SUMFNV PER MESH',F9.6,' IN.W.G.,   ITERATIONS ',I5)
780  FORMAT (/////,1X,'THRESHOLD IN ACCURACY (SUM OF NVP',1X,
     .          'CORRECTIONS PER MESH<2.E-4 IN. W.G.)',/,' NOT ',
     .          'SATISFIED. CURRENT SUMFNV:',F9.6,' IN.W.G., ITER.',I5)
800  FORMAT (/////,12X,'DATA FOR CONCENTRATION AND TEMPERATURE',
     .          ' CALCULATION',//,1X,'AIRWAY  FROM',T16,'TO',T21,
     .          'ELEV. DIFF.',T34,'ROCK T',T43,'CH4 PROD.',T54,'COND.',
     .          T61,'DIFFUSIVITY',/)
810  FORMAT (I5,I7,I5,T21,F9.2,T33,F7.2,T43,F7.1,T52,F7.2,T61,F8.3)
820  FORMAT (///,23X,'DATA FOR FIRE SOURCES',//,1X,'AIRWAY',T11,
     .          'AIRFLOW',T21,'TEMPERATURE',T37,'FUME',T51,'O2 LEFT',T61,
     .          'HEAT INPUT',/)
830  FORMAT (I5,T8,0PF10.0,T22,0PF8.2,T35,2PF6.2,T50,0PF7.2,T62,E8.3)
```

```
840  FORMAT (/,1X,'AIRFLOW REVERSED IN FAN BRANCH NO. ',I5)
890  FORMAT (///,1X,'THE FAILURE IN CONVERGING WAS NOT CAUSED BY A',
     .        ' CYCLIC PATTERN. DATA OF',/,' SIGNIFICANT CONDITION ',
     .        'CHANGE DURING LAST 10 ITERATIONS FOLLOW. FURTHER',/,
     .        ' ANALYSES ARE NEEDED. ("ITR" STANDS FOR ITERATION)',//,
     .        ' UNITS: Q: AIRFLOW RATE IN 1000FT3/MIN;',17X,'T: TEMP. ',
     .        'IN F;',/,8X,'H: NVP CORRECTION PER MESH IN IN.W.G.',//,
     .        ' ITR',T5,I5,9I7,/)
900  FORMAT (///,1X,'THE FAILURE IN CONVERGING WAS CAUSED BY A CYCLIC',
     .        ' PATTERN IN EVERY',I3,/,' ITERATIONS AS SHOWN BELOW:',
     .        ' ("ITR" STANDS FOR ITERATION)',//,' UNITS: Q: AIRFLOW ',
     .        'RATE IN 1000FT3/MIN;',17X,'T: TEMP. IN F;',/,8X,'H: NVP ',
     .        'CORRECTION PER MESH IN IN.W.G.',//,' ITR',T5,I5,9I7,/)
905  FORMAT (1X,'H',T3,10(1X,F6.4),/)
910  FORMAT (1X,'AIRWAY: ',I4,10X,'JS:',I4,6X,'JF:',I4,/,1X,
     .        'T',T3,10F7.1)
920  FORMAT (1X,'Q',T3,10F7.1,/)
930  FORMAT (/,18A4,/)
940  FORMAT (18A4,/)
950  FORMAT (/,18A4)
960  FORMAT (18A4)
970  FORMAT (/////,20X,'TIME AT ',F5.1,' HOUR(S) AFTER EVENT',//)
980  FORMAT (///,T2,'JUNCTION',T12,'TEMP.',T20,'ELEVATION',T32,'CH4',
     .        T40,'JUNCTION',T50,'TEMP.',T58,'ELEVATION',T70,'CH4',/)
990  FORMAT (T2,I5,T10,F7.1,T19,F8.1,T30,F5.2,T40,I5,T48,F7.1,T57,F8.1
     .        ,T68,F5.2)
995  FORMAT (/,'* WARNING *  AIR TEMP. IN FIRE',I5,' IS EXCEPTIONALLY',
     .        ' HIGH FOR MINE FIRE.',/,' THE POSSIBLE CAUSES ARE:',/,
     .        ' 1. "QCENT" WAS NOT SPECIFIED;',/,' 2. "QCENT" DOES NOT',
     .        ' MATCH "HEAT";',/,' 3. "HTPO2" WAS TOO LARGE.')
1040 FORMAT (///,15X,'COEFF. OF POLYNOMIAL OF FAN CURVES',/,20X,
     .        'PF=A0+A1*Q+A2*Q**2+A3*Q**3')
1045 FORMAT (/,T5,'FAN',I5,//,T7,'A0',T19,'A1',T31,'A2',T43,'A3',T52,
     .        'FROM QF',T65,'TO QF',/)
1050 FORMAT (T3,E9.3,T15,E9.3,T27,E9.3,T39,E9.3,T51,E9.3,T63,E9.3)
1060 FORMAT (///,7X,'BASIC DATA TO DEFINE THE NETWORK AND THE ',
     .        'SIMULATION PROBLEM',//,' AIRWAY FROM',T14,'TO',T17,'TY',
     .        T21,'ROCK T',T28,'LENGTH',T35,'ELEV. D',T48,'R',T54,'COND',
     .        T61,'DIFF',T67,'CH4 PR',/)
1070 FORMAT (I5,I6,T12,I4,I3,T20,F6.1,T27,F7.1,T35,F7.1,T44,E8.3,T53,
     .        F5.2,F7.3,F8.1)
1080 FORMAT (/,1X,'* NOTE *  FOR DETAILED DATA LIST, INPUT "IOUT" ',
     .        'AS -1 OR LESS.')
1085 FORMAT (/,2X,'INTERPOLATION:',/,5('  Q*1000   PF '),/)
1090 FORMAT (5(F8.1,F6.2))


C
      END
C
C
C
```

MFIRE2.FOR PROGRAM

```
C
C
C       TRANSIENT-STATE SIMULATION PROGRAM FOR MINE VENTILATION
C                       UNDER THE INFLUENCE OF FIRES
C
C                               VERSION OF JULY 1989
C                                 (PART 2) V 1.20
C
$INCLUDE:'CMMN.DAT'
C
      CHARACTER*50 MD1,MD2,OPTION
C
C
      OPEN (10,FILE='NET',STATUS='UNKNOWN',FORM='UNFORMATTED')
      READ (10) NETWW
      IF (NETWW.EQ.1) GO TO 310
      OPEN (21,FILE='ZHENG.CHG',STATUS='UNKNOWN',FORM='UNFORMATTED')
      READ (21) OPTION
      READ (21) MD1,TACC
      IF (OPTION.EQ.'HUNAN-XIANGYING') GO TO 310
      WRITE (6,760)
      OPEN (20,FILE='LIANG.CHG',STATUS='UNKNOWN',FORM='UNFORMATTED')
      READ (20) NB,NJ,NFNUM,NETW,NTEMP,ITN,DR,TR,TINC,SPAN,IOUT,TOUT,
     .          CONCT,NSTOP
      READ (20) NAV,MAXJ,INFLOW,CRITSM,CRITGS,CRITHT,MNO
      READ (20) TIME,TSTART,JSTART,NSFLOW,WRNPR,WRNGS,WRNSM,WRNHT
      READ (20) MARKX,MARKY,NCOMT1,NCOMTS,NCOMT2,LLINE
      READ (20) JAN,JAJ
      DO 3200 I=1,NB
         READ (20) NO(I),JS(I),JF(I),NWTYP(I),R(I),Q(I),KF(I),LA(I)
         READ (20) A(I),O(I),RSTD(I),DZRD(I),FRNVP(I),NREV(I),RDCH4(I)
         READ (20) RDPROP(I),TRD(I),TJS(I),FFRNVP(I),NNREV(I),JSB(I)
         READ (20) TROCK(I),TMRD(I),HA(I),HK(I)
         READ (20) XNEW(I),DCOAGE(I),CH4V(I),CH4PA(I)
         READ (20) NGIN(I),MMIN(I),NGOUT(I),LOUT(I)
         IF (CONCT.EQ.1) THEN
            QQQ(I)=Q(I)
            TRDD(I)=TRD(I)
            TJSS(I)=TJS(I)
         ENDIF
3200  CONTINUE
      DO 3202 I=1,IMX
         READ (20) NOF(I),NFREG(I),RGRAD(I),NFCW(I),MPTS(I),
     .             NSKP(I),NEGQ(I)
         READ (20) NCENT(I),CONT(I),CONC(I),HEAT(I),O2MIN(I),SMPO2(I),
     .             HTPO2(I),TPR(I),QCENT(I)
         DO 3201 J=1,IMY
            READ (20) QF(I,J),PF(I,J)
            DO 3205 K=1,4
               READ (20) FKQ(I,J,K)
3205        CONTINUE
```

```
3201      CONTINUE
3202 CONTINUE
     DO 3203 I=1,NJ
        READ (20) JNO(I),T(I),Z(I),CH4C(I),JNOL(I),PROP(I),
     .                PRCH4(I),JLR(I)
        IF (CONCT.EQ.1) TTT(I)=T(I)
3203 CONTINUE
     DO 3204 K=1,JAJ
        READ (20) JSTAR(K), TSTAR(K)
3204 CONTINUE
     DO 3206 I=1,MNO
        READ (20) MEND(I),FNVP(I)
3206 CONTINUE
     IYY=MEND(MNO)
     DO 3207 I=1,IYY
        READ (20) MSL(I)
3207 CONTINUE
     DO 3209 I=1,50
        READ (20) ITITLE(I)
        DO 3208 J=1,20
           READ (20) TITLE(I,J)
3208      CONTINUE
3209 CONTINUE
C
     SSP=SPAN*60.0
     REWIND (21)
     WRITE (21) OPTION
     WRITE (21) MD1,SSP
C
1    OPEN (9,FILE=MD1)
     OPEN (8,FILE='YUNZHI.FAN',STATUS='UNKNOWN')
5    IF (LLINE.GT.0) THEN
        READ (9,740) (ROW(I),I=1,80)
        LLINE=LLINE-1
        GO TO 5
     ENDIF
C
C    MARKR: 0: OUTPUT WAS GIVEN IN THE LAST INTERVAL.
C           1: OUTPUT WILL BE OMITTED IN THE PRESENT INTERVAL DUE TO
C           INSIGNIFICANT CHANGE. CHANGES IN Q WILL BE CUMULATED.
C    MH: CONTROLLING THE WRITING OF THE HEADLINE FOR THE CONVERGENCE
C        INFORMATION OF OMITTED OUTPUT.
C    ISQC: COUNTER OF AIRWAYS WHICH HAVE THEIR DETAILED DATA RECORDS.
C
     MARKR=0
     TCUMU=0.0
     IF (TOUT.LT.1.E-5) TOUT=1.0
     MH=0
     MARKX=1
     JDP=0
     TACC=0.0
     ISQC=0
     QS=100.0
     QSUM=0.0
```

```
      WRITE (8,560)
      WRITE (8,550)
      IF (CONCT.EQ.1) WRITE (8,555)
      CALL INIT (1)
C
C     TINC: USER-SPECIFIED SPAN OF TIME INTERVAL IN SEC..
C     QSTD: A CERTAIN PERCENTAGE OF THE SUM OF ABSOLUTE AIRFLOW CHANGES
C           IN AIRWAYS IN THE FIRST INTERVAL.
C     QS: VARIATION IN SUM OF ABS. AIRFLOW CHANGES IN SUCCESSIVE
C         INTERVALS, CRITERION FOR OUTPUT DURING DYNAMIC SIMULATION.
C     DELTAT: TIME INCREMENT THAT IS ACTUALLY ADOPTED IN THE CURRENT
C               SIMULATION UPDATING.
C
      IF (TINC.LE.0.0) TINC=15.0
      DELTAT=TINC
      DO 290 MASTER=1,1000
         IF (MARKR.EQ.0) WRITE (8,560)
         CALL CDCH (NSTOP)
         IF (NSTOP.GT.0) GO TO 310
C
C     MARKP: 0: CONVERGENT RESULTS FOR THE PRESENT INTERVAL HAVE NOT
C              BEEN REACHED.  1: CONVERGENT RESULTS OBTAINED.
C     ITT: COUNTER OF ITERATION IN NETWORK BALANCING.
C     IBB: COUNTER OF ITERATION IN DATA PREPARATION.
C
         MARKP=0
         MEX=0
         ITT=0
         IBB=0
         CALL INIT (2)
C
C     MARKS: TIMER FOR DATA HANDLING.
C     MARKQ: 0: NORMAL CONDITION.  1: SIGNIFICANT CONDITION CHANGES
C              IN JUNCTIONS ARE DETECTED, MORE ITERATIONS ARE NEEDED.
C
160      MARKS=0
C
C     MARKL: 0: NORMAL CONDITION.  1: AIRFLOW REVERSAL FOUND IN THE
C              SYSTEM, RECONSTRUCTION OF MESH FORMATION IS NEEDED.
C
         MARKL=0
         CALL  DTTR (MARKL,MARKP)
         IF (MARKL.NE.0) THEN
            CALL ARR (MARKX)
            CALL BASE
            CALL MSLIST
         ENDIF
         DO 165 I=1,NJ
            TLEFT(I)=-0.1
165      CONTINUE
170      MARKQ=0
         CALL CDJN (ISQC,MARKS)
         DO 180 I=1,NJ
            SUMQ(I)=0.0
```

```
            SUMC(I)=0.0
            SUMM(I)=0.0
            SMHEAT(I)=0.0
180       CONTINUE
          DO 240 I=1,NB
            IQ=IBTN(I)
            IS=NSACB(I,2)
            IF (IBTN(I).NE.0) THEN
              TM=TMRD(I)
              IF (MARKS.GE.1) THEN
                CALL ADPT (IS,IQ,MARKS,I)
              ELSE
                TMST=TACC+DELTAT
                CALL KALPHA (TMST,TM,FX,CP,I)
                COAGE=BI(I)-FX*BI(I)**2/(0.375+BI(I))
                DCOAGE(I)=COAGE
                XNEW(I)=HK(I)*O(I)**2.*COAGE/(120.*DR*QQ(I)*A(I))
                CALL ADPT (IS,IQ,MARKS,I)
              ENDIF
            ELSE
              DO 215 IJ=1,NJ
                KN=JLR(IJ)
                IF(JS(I).EQ.JNO(KN)) THEN
                  DST=QQ(I)*DELTAT/(A(I)*60.0)
                  IF (DST.GT.LA(I)) DST=LA(I)
                  CH4B=(RCH4(I)-PRCH4S(KN))*(LA(I)-DST)/LA(I)+
     .                    PRCH4S(KN)
                  RDCH4(I)=(CH4B*QQ(I)+CH4V(I)*DST/LA(I))/
     .                       (QQ(I)+CH4V(I)*DST/LA(I))
                ENDIF
215           CONTINUE
            ENDIF
            TTB=TRD(I)
            JJ=IABS(JF(I))
            DO 220 J=1,NJ
              IF (JNO(J).EQ.JJ) THEN
                SUMQ(J)=SUMQ(J)+QQ(I)
                SUMC(J)=SUMC(J)+QQ(I)*RDPROP(I)
                SUMM(J)=SUMM(J)+QQ(I)*RDCH4(I)
                SMHEAT(J)=SMHEAT(J)+(0.2376*TTB+1.2E-5*TTB**2)*QQ(I)
                GO TO 240
              ENDIF
220         CONTINUE
240       CONTINUE
          DO 245 I=1,NJ
            JCH(I)=0
245       CONTINUE
C
C
      DO 250 I=1,NJ
        IF (JNOL(I).EQ.JSTART) GO TO 250
C
C       UPDATING THERMAL STATES IN JUNCTIONS EXCEPT FOR THE USER-
C     SPECIFIED STARTING JUNCTION (USUALLY THE ATMOSPHERE).
```

```
C
      K=JLR(I)
      MARKW=0
      TPP=T(K)
      PRPP=PROP(K)
      PRCPP=PRCH4(K)
C
C   PROP(K): CONTENT OF CONTAMINANTS IN AIRFLOW.
C   PRCH4(K): CONTENT OF CH4 IN AIRFLOW.
C   T(K): TEMPERATURE OF AIRFLOW.
C   THE ABOVE PARAMETERS ARE EXPRESSED AS MASS RATIOS WITH RESPECT
C   TO THAT OF AIR CURRENT.
C
      PROP(K)=SUMC(K)/SUMQ(K)
      PRCH4(K)=SUMM(K)/SUMQ(K)
      PTCONT=9.801E7+2.0*SMHEAT(K)/(SUMQ(K)*2.4E-5)
      T(K)=-9900.+SQRT(PTCONT)
      IF (JAN.LE.1) GO TO 248
      DO 247 II=1,JAJ
         IF (K.EQ.JSTAR(II)) THEN
            T(K)=TSTAR(II)
            PROP(K)=0.0
            PRCH4(K)=0.0
         ENDIF
247   CONTINUE
248   IF (CONCT.EQ.1) T(K)=TTT(K)
C
C     MARKW=1 INDICATES THAT ALL THE DOWN-STREAM AIRWAYS OF JUNCTION
C   JNO(K) NEED TO HAVE THEIR DETAILED DATA RECORDS.
C
      IF (ABS(TPP-T(K)).GE.1.0) MARKW=1
      IF (ABS(PRPP-PROP(K)).GE.1.E-3) MARKW=1
      IF (ABS(PRCPP-PRCH4(K)).GE.1.E-3) MARKW=1
      IF (MARKW.EQ.1) JCH(K)=1
      IF (MARKW.EQ.1) MARKQ=1
250 CONTINUE
C
C
      IF (MARKS.LT.1.OR.MARKQ.NE.0) THEN
         IBB=IBB+1
         IF (IBB.LE.7) THEN
            MARKS=MARKS+1
            GO TO 170
         ELSE
            WRITE (8,580) TACC
         ENDIF
      ENDIF
      CALL PARAM
      CALL NVP2
      ITT=ITT+1
      CALL ITR (MARKX,NSFLOW,ITT)
      IF (ITT.GE.(ITN-9)) THEN
         J=ITT-ITN+10
         HRM(J)=SUMFNV/MNO
```

```
            DO 260 I=1,NB
               IF (CONCT.EQ.1) Q(I)=QQQ(I)
               TMRCD(I,J)=TMRD(I)
               QRCD(I,J)=ABS(Q(I))
               IF (JSB(I).NE.IABS(JS(I)).AND.Q(I).GT.0.0)
     .            QRCD(I,J)=-ABS(Q(I))
               IF (JSB(I).EQ.IABS(JS(I)).AND.Q(I).LT.0.0)
     .            QRCD(I,J)=-ABS(Q(I))
260         CONTINUE
         ENDIF
         WRITE (6,770) ITT,SUMFNV/MNO
         IF (ITT.GE.ITN.OR.(SUMFNV/MNO.LE.2.E-4.AND.ITT.GE.2)) THEN
            CALL EVHT (MARKX,MARKP,MARKL,MARKR,QSUM,QS)
         ELSE
            IF (ITT.GE.(ITN-2).AND.IOUT.LT.0) THEN
               WRITE (8,720) ITT
               TACC=TACC+TINC
               CALL OUTPUT (4,MARKX,MARKY,ITT)
               IF (IOUT.LE.(-2)) CALL OUTPUT (6,MARKX,MARKY,ITT)
               TACC=TACC-TINC
               WRITE (8,730)
            ENDIF
            CALL DISP (MARKX,ISQC,IBB)
            GO TO 160
         ENDIF
         IF (MASTER.LE.1) QSTD=QSUM/7.
         WRITE (6,530) TACC
         TCUMU=TCUMU+DELTAT
         IF (TACC.LT.(SPAN*60.)) THEN
            IF (QSUM.LE.QSTD.AND.ITT.LT.ITN.AND.TCUMU.LT.TOUT*60.) THEN
               MARKR=MARKR+1
               TACC1=TACC/60.0
               IF (MH.EQ.0) THEN
                  MH=1
                  WRITE (8,590)
               ENDIF
               WRITE (8,600) SUMFNV/MNO,ITT,TACC1
            ELSE
               MARKR=0
               TCUMU=0.0
               MH=0
               CALL OUTPUT (4,MARKX,MARKY,ITT)
               IF (IOUT.LE.(-2)) CALL OUTPUT (6,MARKX,MARKY,ITT)
               IF (IOUT.NE.0) CALL OUTPUT (5,MARKX,MARKY,ITT)
            ENDIF
         ELSE
            CALL OUTPUT (4,MARKX,MARKY,ITT)
            IF (IOUT.LE.(-2)) CALL OUTPUT (6,MARKX,MARKY,ITT)
            IF (IOUT.NE.0) CALL OUTPUT (5,MARKX,MARKY,ITT)
            GO TO 300
         ENDIF
290   CONTINUE
C
300   WRITE (6,640)
```

```
310  STOP
C
530  FORMAT (/,1X,'*** SIMULATION GOES ON, ',F7.0,' SEC. PASSED.',/)
550  FORMAT (T19,'NON-STEADY STATE SIMULATION OUTPUT')
555  FORMAT (T16,'---FOR CONCENTRATION CALCULATION ONLY---')
560  FORMAT (///,1X,71('*'),///)
580  FORMAT (//,1X,'* ATTENTION *  SIMULATION STEP DELTAT NEEDS TO ',
     .            'BE SHORTENED AT',/,' ABOUT',F7.0,' SEC. AFTER EVENT ',
     .            'IF DETAILED ANALYSIS IS WANTED.')
590  FORMAT (//,1X,'* ATTENTION *   OUTPUT FOR  THE FOLLOWING TIME ',
     .           'INCREMENTS WERE',/,1X,'OMITTED DUE TO INSIGNIFICANT ',
     .           'CONDITION CHANGES ( THE THRESHOLD ',/,1X,'IN ACCURACY: ',
     .           'NVP CORR. PER MESH < 0.0002 IN.W.G. )',//,5X,'SUM OF ',
     .           'NVP CORR',T30,'ITERATIONS',T45,'AT TIME IN MIN.',/)
600  FORMAT (7X,F9.5,T31,I5,T47,F6.2)
630  FORMAT (///,1X,71('*'),///,T14,'QUASI-EQUILIBRIUM STATE',
     .           ' SIMULATION OUTPUT',///,1X,71('*'))
640  FORMAT (/,1X,'DYNAMIC SIMULATION PART COMPLETED.',//)
720  FORMAT (//,1X,71('-'),///,' * ATTENTION *  THE FOLLOWING ',
     .           'IS OUTPUT FOR ITERATION ',I4,/,' "DELTA Q" IS ',
     .           'MEANINGLESS HERE.',/)
730  FORMAT (//,1X,71('-'))
740  FORMAT (80A1)
760  FORMAT (/,1X,'ENTERING DYNAMIC SIMULATION PART OF THE PROGRAM.',/
     .           ,1X,'PREFIXED ACCURACY CRITERION = 0.0002 IN.W.G. PER ',
     .           'MESH.',/)
770  FORMAT (10X,'ITERATION ',I3,'  COMPLETED, CURRENT ERROR = ',
     .           F8.5)
C
     END
C
C
     BLOCK DATA
C
C    ------------------------------------------------------------------------------------------------------------------------
C
C       SUBPROGRAM PURPOSE:
C       INITIATION OF THE VARIABLES AND ARRAYS IN COMMON BLOCKS.
C
C    ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
     PARAMETER ( K1=NMX*4,  K2=NMX*3,  K3=NMX*6,  K4=NMX*8,  K5=NMY*3,
     .           K6=IMX*2,   K7=IMX*IMY*2,      K8=LMX+NMY+1,   K9=NMY+NMX,
     .           K10=NMY*2, K11=NMX*2+IMX*11,   K16=NMZ*10*5,     K22=NMX*2,
     .           K17=NMZ*10*5+NMX*2, K19=NMZ*22, K21=IMX*3,       K23=NMY*4,
     .           K24=IMY*IMY+IMY*2+IMX*IMY*4 )
C
C
     DATA  NO,JS,JF,NWTYP,R,Q,P,KF,LA,A,O,RSTD,DZRD,FRNVP,NREV,RDCH4,
     .     RDPROP,TRD,TJS,RDOP,RCH4,FFRNVP,RA,NNREV
     .     /K1*0,K2*0.0,NMX*0,K3*0.0,NMX*0,K4*0.0,NMX*0/
     DATA  NOF,NFREG,RGRAD,NFCW,MPTS,QF,PF,NSKP,NEGQ,JSB
     .     /K6*0,IMX*0.0,K6*0,K7*0.0,IMX*0,IMX*0,NMX*0/
```

```
      DATA  FC,DK,FK,FKQ
     .      /K24*0.0/
      DATA  MNO,MEND,MSL,FNVP,RQ,INU,KNO,KJS,KJF
     .      /K8*0,K9*0.0,K1*0/
      DATA  JNO,T,Z,CH4C,JNOL,PROP,PRCH4,JLR
     .      /NMY*0,K5*0.0,NMY*0,K10*0.0,NMY*0/
      DATA  TROCK,TMRD,HA,HK
     .      /K1*0.0/
      DATA  NCENT,CH4V,CH4PA,CONT,CONC,HEAT,O2MIN,SMPO2,HTPO2,TFSI,
     .      O2BEH,TPR,HTAD,QCENT
     .      /NMX*0,K11*0.0/
      DATA  NGIN,MMIN,NGOUT,LOUT
     .      /K1*0/
      DATA  MEMREC,NOREC,ESTPR,ESTCH4,ESTTR
     .      /K22*0,K2*0.0/
      DATA  XNEW,DCOAGE
     .      /K22*0.0/
      DATA  TIME,TSTART,JSTART,TLEFT
     .      /0.0,0.0,0,NMX*0.0/
      DATA  SUMQ,SUMC,SUMM,SMHEAT
     .      /K23*0.0/
      DATA  PROPS,PRCH4S,QQ,QTP,TTJS,TTRD,TAS,BI
     .      /K4*0.0/
      DATA  DELTAT,TACC,SUMFNV
     .      /3*0.0/
      DATA  IBTN,DPPA,NSAC,TMRDA,DELTAQ,DPPB,NSACB,GBTN,JCH,TAUXA,
     .      TAUXB,TAUXC,TAUXD,FMASS
     .      /NMX*0,K16*0.0,K22*0,K17*0.0,K22*0,50*0.0,NMY*0,K19*0.0,
     .      NMZ*0.0/
      DATA  WRNPR,WRNGS,WRNSM,WRNHT,IOMIT,ROMIT,IAC
     .      /4*0.0,IMX*0,K21*0.0,0/
      DATA  ITITLE,NCOMT1,NCOMTS,NCOMT2,LLINE
     .      /54*0/
C
      END
C
C
      SUBROUTINE ARR (MARKX)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSES:
C        1) GROUPING AIRWAYS ACCORDING TO THEIR TYPES.
C        2) ARRANGING REGULAR AIRWAYS ACCORDING TO THE PRODUCT OF
C           Q AND R.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
      NBU=NB
      NBL=1
      DO 10 K=1,NB
         IF (NWTYP(K).LT.0) THEN
```

```
            INU(NBU)=K
            NBU=NBU-1
          ELSE IF (NWTYP(K).EQ.0.OR.NWTYP(K).EQ.10) THEN
            RQ(K)=ABS(R(K)*Q(K))
            NWTYP(K)=2
          ELSE
            IF (NFNUM.GT.0) THEN
              DO 5 J=1,NFNUM
                IF (NOF(J).EQ.NO(K)) GO TO 10
5             CONTINUE
            ENDIF
            INU(NBL)=K
            NBL=NBL+1
          ENDIF
10      CONTINUE
        IF (NFNUM.GT.0) THEN
          DO 25 K=1,NB
            DO 20 J=1,NFNUM
              IF (NOF(J).EQ.NO(K)) THEN
                RGRAD(J)=0.0
                NFCW(J)=0
                NFREG(J)=K
                INU(NBU)=K
                NBU=NBU-1
                GO TO 25
              ENDIF
20          CONTINUE
25        CONTINUE
        ENDIF
C
C
        IF (NBU.GT.(NBL-1)) THEN
30        NRETU=0
          DO 40 K=1,NB
            IF (NWTYP(K).GE.2) THEN
              IF (NRETU.LE.0) THEN
                NMIN=K
                NRETU=1
              ENDIF
              IF (RQ(NMIN).GT.RQ(K)) NMIN=K
            ENDIF
40        CONTINUE
          INU(NBL)=NMIN
          NBL=NBL+1
          NWTYP(NMIN)=0
          IF ((NBU+1).GT.NBL) GO TO 30
        ENDIF
        IF (MARKX.GT.0) THEN
          DO 60 I=1,INFLOW
            DO 50 J=1,NB
              IF (NCENT(I).EQ.NO(J)) THEN
                NWTYP(J)=10
                GO TO 60
              ENDIF
```

```
50          CONTINUE
60        CONTINUE
      ENDIF
C
      RETURN
      END
C
C
      SUBROUTINE BASE
C
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       FORMATION OF BASE SYSTEM.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
      IND=INU(1)
      KJF(NJ)=JS(IND)
      KJS(NJ-1)=JS(IND)
      JS(IND)=-JS(IND)
      KJF(NJ-1)=JF(IND)
      KNO(NJ-1)=IND
      KNUM=NJ-1
10    DO 30 NUC=2,NB
        IND=INU(NUC)
        IF (JS(IND).LT.0) GO TO 30
        N1=0
        N0=0
        DO 20 K=KNUM,NJ
          IF (JS(IND).EQ.KJF(K)) N0=1
          IF (JF(IND).EQ.KJF(K)) N1=1
20      CONTINUE
        IF ((N1+N0).EQ.1) THEN
          IF (NWTYP(IND).LT.0) WRITE (8,100) NO(IND)
          KNUM=KNUM-1
          IF (N0.LE.0) THEN
            KJS(KNUM)=JF(IND)
            KJF(KNUM)=JS(IND)
            KNO(KNUM)=-IND
            JS(IND)=-JS(IND)
          ELSE
            KJS(KNUM)=JS(IND)
            KJF(KNUM)=JF(IND)
            KNO(KNUM)=IND
            JS(IND)=-JS(IND)
          ENDIF
          GO TO 10
        ELSE IF ((N1+N0).GT.1) THEN
          JS(IND)=-JS(IND)
          JF(IND)=-JF(IND)
```

```
        ENDIF
30    CONTINUE
      DO 40 K=1,NB
         JS(K)=IABS(JS(K))
40    CONTINUE
C
100   FORMAT (/,1X,'* ATTENTION *  REGULATOR',I5,'IN BASE SYSTEM')
C
      RETURN
      END
C
C
      SUBROUTINE MSLIST
C
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSES:
C        SET-UP OF MESH LIST.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
      MESC=0
      MNO=0
      DO 60 K=1,NB
         IF (JF(K).LT.0) THEN
           JF(K)=-JF(K)
           MNO=MNO+1
           JBM=JS(K)
           JEM=JF(K)
           NK=K
10         MESC=MESC+1
           MEND(MNO)=MESC
           MSL(MESC)=NK
           DO 20 KCO=1,NJ
              IF (JBM.EQ.KJF(KCO)) THEN
                KB=KCO
                GO TO 30
              ENDIF
20         CONTINUE
30         DO 40 KCO=1,NJ
              IF (JEM.EQ.KJF(KCO)) THEN
                KE=KCO
                GO TO 50
              ENDIF
40         CONTINUE
50         IF (KB.LT.KE) THEN
             NK=KNO(KB)
             JBM=KJS(KB)
             GO TO 10
           ELSE IF (KB.GT.KE) THEN
             NK=-KNO(KE)
```

```
            JEM=KJS(KE)
            GO TO 10
          ENDIF
        ENDIF
60    CONTINUE
      IF (MEND(MNO).GT.LMX) WRITE (8,100)
C
      RETURN
100   FORMAT (/,1X,'* ERROR *  NETWORK TOO BIG, INCREASE LMX.')
C
      END
C
C
      SUBROUTINE ITR (MARKX,NSFLOW,ITT)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       APPLICATION OF HARDY CROSS METHOD TO NETWORK BALANCING.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
5     IT=0
10    DQSUM=0.
      MBEGW=1
      DO 50 K=1,MNO
        MENDW=MEND(K)
        DPSUM=0.
        RQSUM=0.
        N=MSL(MBEGW)
        IF (N.LT.0) N=-N
        IF (NWTYP(N).EQ.(-1)) THEN
          MBEGW=MENDW+1
          GO TO 50
        ENDIF
        DO 30 J=MBEGW,MENDW
          N=MSL(J)
          FACT=1.
          IF (N.LT.0) THEN
            N=-N
            FACT=-1.
          ENDIF
          IF(NWTYP(N).EQ.1) THEN
            IF (NFNUM.GT.0) THEN
              DO 20 L=1,NFNUM
                IF (NFREG(L).EQ.N) THEN
                  RQSUM=RQSUM-RGRAD(L)
                  GO TO 25
                ENDIF
20            CONTINUE
            ENDIF
25          IF (IABS(JS(N)).NE.JSB(N)) FACT=-FACT
```

```
                DPSUM=DPSUM-FACT*R(N)
              ELSE
                RQ2=R(N)*ABS(Q(N))*2.E-5
                RQSUM=RQSUM+RQ2
                DP=R(N)*Q(N)*ABS(Q(N))*1.E-10
                DPSUM=DPSUM+FACT*DP
              ENDIF
30          CONTINUE
            DQ=(DPSUM-FNVP(K))*1.E5/RQSUM
            DO 40 J=MBEGW,MENDW
              N=MSL(J)
              FACT=1.
              IF (N.LT.0) THEN
                N=-N
                FACT=-1.
              ENDIF
              Q(N)=Q(N)-(DQ*FACT)
40          CONTINUE
            DQSUM=DQSUM+ABS(DQ)
            MBEGW=MENDW+1
50        CONTINUE
C
          DO 100 KI=1,NB
            IF (NFNUM.GT.0.AND.NWTYP(KI).EQ.1) THEN
              DO 90 J=1,NFNUM
                IF (NOF(J).EQ.NO(KI)) THEN
                  NFCW(J)=0
                  NEGQ(J)=0
                  IF (Q(KI).LT.0.0.AND.IABS(JS(KI)).EQ.JSB(KI))
     .              NEGQ(J)=1
                  IF (Q(KI).GT.0.0.AND.IABS(JS(KI)).NE.JSB(KI))
     .              NEGQ(J)=1
                  NPTS=MPTS(J)
                  NABF=JS(KI)
                  DO 70 L=1,NJ
                    IF (NABF.EQ.JNO(L)) THEN
                      TABF=T(L)
                      GO TO 80
                    ENDIF
70                CONTINUE
80                IF (NSKP(J).EQ.0) THEN
                    CALL SPLINE (J,Q(KI),R(KI),TABF,0)
                    NSKP(J)=1
                  ELSE
                    CALL SPLINE (J,Q(KI),R(KI),TABF,1)
                  ENDIF
                  GO TO 100
                ENDIF
90            CONTINUE
            ENDIF
100       CONTINUE
C
          IT=IT+1
          IF ((DQSUM/FLOAT(MNO)).LT.10.) RETURN
```

```
      IF (IT.GT.100) THEN
         WRITE (8,210) DQSUM/FLOAT(MNO),ITT
         RETURN
      ENDIF
      GO TO 10
C
200   FORMAT (//,1X,'* ATTENTION *  ACCURACY CRITERION IN NETWORK',
     .        ' BALANCING',/,1X,'( SUB. ITR ) WAS NOT SATISFIED IN 100'
     .        ,'ITERATIONS. THE',/,' CRITERION: 10 PER MESH,',6X,
     .        ' CURRENT ERROR:',F8.0)
210   FORMAT (//,1X,'* ATTENTION *  ACCURACY CRITERION IN NETWORK',
     .        ' BALANCING',/,1X,'( SUB. ITR ) WAS NOT SATISFIED IN 100'
     .        ,'ITERATIONS. THE',/,' CRITERION: 10 PER MESH,',6X,
     .        ' CURRENT ERROR:',F8.0,/,' IT WAS HAPPENED IN DATA ',
     .        'PREPARATION CYCLE ',I5)
C
      END
C
C
      SUBROUTINE SPLINE (J,QKI,RKI,TABF,NTW)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       1) FAN CURVE FITTING.
C       2) FAN PRESSURE INTERPOLATION.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
C     FKQ(J,I,1-4): COEFFICIENTS. OF 3RD ORDER POLYNOMIAL FOR FAN NOF(J),
C                   SEGMENT I.
C     NSKP(J)=1: MARKER INDICATING THAT FAN CURVE FITTING FOR FAN
C                 NOF(J) HAS BEEN PROCESSED.
C
      MF = MPTS(J)
      IF (NTW.EQ.0) THEN
         DELQ=1.0
         DELP=0.0
         DQP=0.0
         MFA = MF-1
         DO 10 I = 1,MFA
            FK(I) = 0.0
            DK(I) = 0.0
            DO 10 K = 1,MFA
               FC(I,K) = 0.0
10       CONTINUE
         FC(1,1) = 2.0
         FC(1,2) = 1.0
         FC(MFA,MFA-1) = 1.0
         FC(MFA,MFA) = 2.0
         DO 20 I = 1,MFA
            DQA = DELQ
```

```
            DELQ = QF(J,I+1)-QF(J,I)
            DPA = DELP
            DELP = PF(J,I+1)-PF(J,I)
            DQPA = DQP
            DQP = DELP/DELQ
            IF (I.GT.1.AND.I.LT.MFA) THEN
               DK(I) = (DQPA*DELQ+DQP*DQA)*3.0
               FC(I,I-1) = DELQ
               FC(I,I) = 2.0*(DQA+DELQ)
               FC(I,I+1) = DQA
            ELSE
               DK(I) = DQP*3.0
            ENDIF
20       CONTINUE
         DO 25 I = 2,MFA
            FC(I,I) = FC(I,I)-FC(I-1,I)*FC(I,I-1)/FC(I-1,I-1)
            DK(I) = DK(I)-DK(I-1)*FC(I,I-1)/FC(I-1,I-1)
25       CONTINUE
         FK(MFA) = DK(MFA)/FC(MFA,MFA)
C
C
         DO 30 I = 2,MFA
            II = MFA-I+1
            FK(II) = (DK(II)-FC(II,II+1)*FK(II+1))/FC(II,II)
30       CONTINUE
         DO 35 I = 1,MFA
            FXD = QF(J,I+1)-QF(J,I)
            FXA = 1.0/FXD**2.
            FXB = FK(I)-(PF(J,I+1)-PF(J,I))/FXD
            FXC = FK(I+1)-(PF(J,I+1)-PF(J,I))/FXD
            FKQ(J,I,1) = (QF(J,I+1)*PF(J,I)-QF(J,I)*PF(J,I+1))
     .                   /FXD-FXA*(FXB*QF(J,I)*QF(J,I+1)**2+
     .                   FXC*QF(J,I+1)*QF(J,I)**2)
            FKQ(J,I,2) = (PF(J,I+1)-PF(J,I))/FXD+FXA*(FXB*
     .                   (QF(J,I+1)**2+2.*QF(J,I+1)*QF(J,I))+
     .                   FXC*(QF(J,I)**2+2.*QF(J,I+1)*QF(J,I)))
            FKQ(J,I,3) = -FXA*(FXB*(2.*QF(J,I+1)+QF(J,I))+FXC*
     .                   (QF(J,I+1)+2.*QF(J,I)))
            FKQ(J,I,4) = FXA*(FXB+FXC)
35       CONTINUE
         NSKP(J)=1
      ENDIF
C
C
      QKI=ABS(QKI)
      IF (QKI.LE.QF(J,1).OR.NEGQ(J).GE.1) THEN
         RKI = PF(J,1)*(TR+460.)/(TABF+460.)
         RGRAD(J) = 0.0
         NFCW(J) = 1
         RETURN
      ELSE IF (QKI.GE.QF(J,MF)) THEN
         RKI = PF(J,MF)*(TR+460.)/(TABF+460.)
         RGRAD(J) = 0.0
         NFCW(J) = 1
```

```
      RETURN
    ENDIF
    IF (QF(J,1).GE.QKI) THEN
      M = 1
    ELSE
      DO 40 K = 2,MF
        IF (QF(J,K).GE.QKI) THEN
          M = K-1
          GO TO 45
        ENDIF
40    CONTINUE
    ENDIF
    M = MFA
45  RKI = FKQ(J,M,1)+FKQ(J,M,2)*QKI+FKQ(J,M,3)*QKI**2
     .          +FKQ(J,M,4)*QKI**3
    RKI=RKI*(TR+460.)/(TABF+460.)
    RGRAD(J) = FKQ(J,M,2)+2*FKQ(J,M,3)*QKI+
     .                3*FKQ(J,M,4)*QKI**2
C
    RETURN
    END

C
C
    SUBROUTINE  KALPHA (TMST,TM,FX,CP,I)
C
C   ------------------------------------------------------------------------------------------------------------------------
C
C     SUBROUTINE PURPOSE:
C     EVALUATION OF "COEFFICIENT OF AGE" ASSUMING DRY CASE.
C
C   ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
C   WT: TEMP. CORRECTED AIR DENSITY, LBM/FT**3.
C   VISC: KINETIC VISCOSITY OF AIR, FT/HR.
C   HKA: THERMAL CONDUCTIVITY OF ROCK, BTU/HR*FT*F.
C   FO: FOURIER NUMBER.  BI: BIOT NUMBER.  RN: REYNOLDS NUMBER.
C   HC: HEAT TRANSFER COEFFICIENT.
C
    PI=3.1415936
    CP=0.24
    TMX=TM
    QX=QQ(I)
    TX=TMST/3600.
    WT=DR*(TR+460.)/(TMX+460.)
    VISC=0.000145*((460.+TMX)/492.)**1.75
    HKA=0.014*((460.+TMX)/492.)**0.81
    RN=QX*DR/(15.*WT*O(I)*VISC)
    FR0=0.0032+0.221/RN**0.237
    POT=(100./RN)**0.125
    COR=((KF(I)*0.075)/(809*DR*FR0))**POT
    HC=0.005*HKA*O(I)*RN**0.8*COR/A(I)
```

```
      FO=TX*HA(I)*O(I)**2/(4.*A(I)**2)
      BI(I)=HC*2.*A(I)/(O(I)*HK(I))
      X=(0.375+BI(I))*SQRT(FO)
C
      IF (X.LT.2.5) THEN
         N=0
         SUMT=0.0
         ADDT=X
10       SUMT=SUMT+ADDT
         N=N+1
         ADDT=-ADDT*X**2*(2.*N-1)/(N*(2.*N+1))
         IF (ABS(ADDT).GE.0.00001) GO TO 10
         FX=1.0-(EXP(X**2))*(1.0-(2.0/SQRT(PI))*SUMT)
      ELSE
         N=0
         SUMT=0.0
         ADDT=1.0
20       SUMT=SUMT+ADDT
         N=N+1
         OLADDT=ADDT
         ADDT=-ADDT*(2*N-1)/(2.0*X**2)
         IF (ABS(ADDT).LT.0.00001) GO TO 30
         IF (ABS(OLADDT).GT.ABS(ADDT)) GO TO 20
30       FX=1.0-1.0*SUMT/(SQRT(PI)*X)
      ENDIF
C
      RETURN
      END
C
C
      SUBROUTINE  NVP2
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSE:
C        EVALUATION OF NVP ACCORDING TO CALCULATED MEAN AIR TEMP. IN
C     AIRWAYS (REFER TO SUBS. TEVAL AND PARAM).
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
      MBEGW=1
      SUMFNV=0.0
      DO 20 K=1,MNO
          MENDW=MEND(K)
C
C     FNTM: AVERAGE TEMPERATURE IN THE MESH.
C     FNVP(K): NVP IN MESH K. IN IN.W.G.
C     SUMFNV: SUM OF NVP VARIATION IN MESHES FROM ITER. TO ITER.
C
          ONVP=FNVP(K)
          FNVP(K)=0.
          ZUP=0.0
```

```
            ZDOWN=0.0
            HSU=0.
            TSU=0.
            DO 10 J=MBEGW,MENDW
               N=MSL(J)
               IF (N.LT.0) THEN
                  FACT=-1.
                  NX=-N
               ELSE
                  FACT=1.
                  NX=N
               ENDIF
               HSU=HSU+FACT*FRNVP(NX)
               TSU=TSU+FACT*FFRNVP(NX)
10          CONTINUE
            IF (ABS(TSU).GT.1.E-3.AND.ABS(HSU).GT.1.E-5) THEN
               HTU=ABS(HSU/TSU)
               FNTM=SQRT(HTU)
               FNVP(K)=HSU*DR/(5.2*FNTM)
            ELSE
               FNVP(K)=0.0
            ENDIF
            SUMFNV=SUMFNV+ABS(ONVP-FNVP(K))
            MBEGW=MENDW+1
20       CONTINUE
C
         RETURN
         END
C
C
         SUBROUTINE  INIT (IPT)
C
C        ------------------------------------------------------------------------------------------------------------------------
C
C           SUBROUTINE PURPOSE:
C           INITIATION OF DATA STORAGE ARRAYS BEFORE DYNAMIC SIMULATION OR
C        BEFORE EACH TIME INCREMENT.
C
C        ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
         IF (IPT.EQ.1) THEN
            DO 30 I=1,NB
               RA(I)=R(I)
               TMRDA(I)=TMRD(I)
               IF (JS(I).LT.0) JS(I)=-JS(I)
               IF (JF(I).LT.0) JF(I)=-JF(I)
30          CONTINUE
            DO 50 I=1,INFLOW
               DO 40 J=1,NB
                  IF (NCENT(I).NE.NO(J)) GO TO 40
                  NWTYP(J)=10
                  GO TO 50
```

```
40          CONTINUE
50        CONTINUE
      ELSE
C
          DO 70 I=1,NJ
              TAS(I)=T(I)
              PROPS(I)=PROP(I)
              PRCH4S(I)=PRCH4(I)
70        CONTINUE
          DO 80 I=1,NMZ
              FMASS(I)=-1.0
80        CONTINUE
          DO 90 I=1,INFLOW
              HTAD(I)=0.0
90        CONTINUE
          DO 100 I=1,NB
              IF (NNREV(I).EQ.(NNREV(I)/2)*2) THEN
                 NREV(I)=0
              ELSE
                 NREV(I)=1
              ENDIF
              TTJS(I)=TJS(I)
              TTRD(I)=TRD(I)
              RDOP(I)=RDPROP(I)
              RCH4(I)=RDCH4(I)
              QQ(I)=Q(I)
              QTP(I)=Q(I)
              IF (CONCT.EQ.1) THEN
                 TTJS(I)=TJSS(I)
                 TTRD(I)=TRDD(I)
                 QQ(I)=QQQ(I)
                 QTP(I)=QQQ(I)
              ENDIF
100       CONTINUE
      ENDIF
C
      RETURN
      END
C
C
      SUBROUTINE CDCH (NSTOP)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C         SUBROUTINE PURPOSE:
C     UPDATING CONDITION CHANGES ACCORDING TO USER-SPECIFIED TIME TABLE
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
      DIMENSION  QFX(20),PFX(20),DAL(15)
C
C
```

```
      IF (TACC.LT.DELTAT) IREAD=1
C
C     ICODE: CODE NUMBER. 1: CHANGE TO ORDINARY AIRWAY.  2: CHANGE
C            TO FAN BRANCH.  3: CHANGE TO FIRE SOURCE. 4: FIRE CEASED.
C            5: CHANGE SIMULATION STEPS. 6: JUNCTION WHOSE DOWN-STREAM
C            SIDE AIRWAYS HAVE DETAILED DATA RECORDS.
C     NBR: AIRWAY CALLING NUMBER IN WHICH CONDITION CHANGE HAPPENS.
C     INPUT SCHEME:
C       TS,ICODE,(MAY BE FOLLOWED BY OTHER PARAMETERS)
C       TS,1,NBR,RSTD
C       TS,2,NBR,INDEX,QF1,PF1,QF2,PF2,QF3,PF3,QF4,PF4,QF5,PF5
C       TS,3,NBR,CONT,CONC,HEAT,O2MIN,SMPO2,HTPO2,QCENT,TPR
C       TS,4,NBR
C       TS,5,DELTAT
C       TS,6,TOUT
C       TS,7,JDPP
C
1     IF (IREAD.EQ.1) THEN
4       CALL READIN (DAL,14,ISTOP,0)
        IF (ISTOP.EQ.1) THEN
          ICODE=DAL(2)
          WRITE (8,300)
          IF (ICODE.EQ.1) WRITE (8,310)
          IF (ICODE.EQ.2) WRITE (8,320)
          IF (ICODE.EQ.3) WRITE (8,330)
          IF (ICODE.EQ.4) WRITE (8,340)
          IF (ICODE.EQ.5) WRITE (8,350)
          IF (ICODE.EQ.6) WRITE (8,360)
          IF (ICODE.EQ.7) WRITE (8,365)
          WRITE (8,370) (ROW(IE),IE=1,80)
          NSTOP=5
          RETURN
        ENDIF
        IF (DAL(15).LT.(-1.E20)) GO TO 4
        IF (NCOMTS.GT.NCOMT2) THEN
          NCOMT2=NCOMTS
          GO TO 4
        ENDIF
        IREAD=0
        TS=DAL(1)
      ENDIF
      IF (TS*60.0.GT.TACC) RETURN
      ICODE=DAL(2)
C
      IF (ICODE.EQ.5) THEN
        DELT=DAL(3)
        IF (DELT.GT.1.E-3.AND.DELT.LT.30.0) THEN
          WRITE (8,190) DELTAT,DELT,TS
          DELTAT=DELT
        ELSE
          WRITE (8,200) DELT
        ENDIF
        GO TO 100
C
```

```
      ELSE IF (ICODE.EQ.6) THEN
         TOUT=DAL(3)
         IF (TOUT.LT.1.E-5) TOUT=1.0
         GO TO 100
C
      ELSE IF (ICODE.EQ.7) THEN
         JDP=JDP+1
         IF (JDP.GT.IMX) THEN
            WRITE (8,240)
            JDP=JDP-1
            GO TO 100
         ENDIF
         JDPP(JDP)=DAL(3)
         GO TO 100
      ENDIF
      NBR=DAL(3)
      DO 5 I=1,NB
          IF (NO(I).EQ.NBR) THEN
             N=I
             GO TO 6
          ENDIF
5     CONTINUE
      WRITE (8,210) NBR
      GO TO 100
C
C       ICODE=1 INDICATES THAT AIRWAY NBR BECOMES AN ORDINARY AIRWAY
C     WITH RESISTANCE EQUAL TO RSTD AT TS MIN. AFTER EVENT.
C
6     IF (ICODE.EQ.1) THEN
         RSTD(N)=DAL(4)
         R(N)=RSTD(N)*(TMRD(N)+460.0)**2/(TR+460.0)**2
         NWTYP(N)=0
         IF (NFNUM.GT.0) THEN
            DO 8 J=1,NFNUM
                IF (NOF(J).EQ.NBR) THEN
                   IF (NFNUM.GT.J) THEN
                      DO 7 K=J,NFNUM-1
                          NOF(K)=NOF(K+1)
                          MPTS(K)=MPTS(K+1)
                          NSKP(K)=NSKP(K+1)
7                     CONTINUE
                   ENDIF
                   NOF(NFNUM)=0
                   MPTS(NFNUM)=0
                   NSKP(NFNUM)=0
                   NFNUM=NFNUM-1
                   GO TO 9
                ENDIF
8           CONTINUE
         ENDIF
9        WRITE (8,110) NO(N),RSTD(N),TS
C
      ELSE IF (ICODE.EQ.2) THEN
         INDEX=DAL(4)
```

```
      IF (INDEX.LE.0) THEN
        WRITE (8,120) INDEX,NBR
        GO TO 100
      ELSE IF (INDEX.EQ.1) THEN
        RX=DAL(5)
        IF (NFNUM.GT.0) THEN
          DO 50 J=1,NFNUM
            IF (NOF(J).EQ.NBR) THEN
              DO 40 JJ=J,NFNUM-1
                NOF(JJ)=NOF(JJ+1)
                MPTS(JJ)=MPTS(JJ+1)
                DO 40 KK=1,IMX
                  QF(JJ,KK)=QF(JJ+1,KK)
                  PF(JJ,KK)=PF(JJ+1,KK)
40            CONTINUE
              NFNUM=NFNUM-1
            ENDIF
50        CONTINUE
        ENDIF
        NWTYP(N)=1
        R(N)=RX
        WRITE (8,150) NBR,R(N)
        GO TO 100
      ELSE IF (INDEX.GT.5) THEN
        WRITE (8,120) INDEX,NBR
        INDEX=5
      ENDIF
      DO 55 I=1,INDEX
        II=5+(I-1)*2
        QFX(I)=DAL(II)
        PFX(I)=DAL(II+1)
55    CONTINUE
      IF (NFNUM.GT.0) THEN
        DO 20 J=1,NFNUM
          IF (NOF(J).EQ.NO(N)) THEN
            MPTS(J)=INDEX
            DO 10 K=1,INDEX
              QF(J,K)=QFX(K)
              PF(J,K)=PFX(K)
10          CONTINUE
            JN=J
            GO TO 35
          ENDIF
20      CONTINUE
      ENDIF
      NFNUM=NFNUM+1
      IF (NFNUM.GT.IMX) THEN
        WRITE (8,130) NO(N),PFX(1)
        NFNUM=NFNUM-1
        NWTYP(N)=1
        R(N)=PFX(1)
        GO TO 100
      ENDIF
      NWTYP(N)=1
```

```
      NOF(NFNUM)=NO(N)
      MPTS(NFNUM)=INDEX
      DO 30 K=1,INDEX
         QF(NFNUM,K)=QFX(K)
         PF(NFNUM,K)=PFX(K)
30    CONTINUE
      JN=NFNUM
35    WRITE (8,140) NBR,TS,(QF(JN,K),PF(JN,K),K=1,INDEX)
C
    ELSE IF (ICODE.EQ.3) THEN
      CONTX=DAL(4)
      CONCX=DAL(5)
      HEATX=DAL(6)
      O2MINX=DAL(7)
      SMPO2X=DAL(8)
      HTPO2X=DAL(9)
      QCENTX=DAL(10)
      TPRX=DAL(11)
      IF (TPRX.LT.1.E-5) TPRX=1.E-5
      IF (INFLOW.GT.0) THEN
        DO 60 J=1,INFLOW
          IF (NCENT(J).EQ.NBR) THEN
            CONT(J)=CONTX
            CONC(J)=CONCX
            HEAT(J)=HEATX
            O2MIN(J)=O2MINX
            SMPO2(J)=SMPO2X
            HTPO2(J)=HTPO2X
            QCENT(J)=QCENTX
            TPR(J)=TPRX
            JN=J
            GO TO 70
          ENDIF
60      CONTINUE
      ENDIF
      INFLOW=INFLOW+1
      IF (INFLOW.GT.IMX) THEN
        WRITE (8,260) NBR
        GO TO 100
      ENDIF
      NCENT(INFLOW)=NBR
      CONT(INFLOW)=CONTX
      CONC(INFLOW)=CONCX
      HEAT(INFLOW)=HEATX
      O2MIN(INFLOW)=O2MINX
      SMPO2(INFLOW)=SMPO2X
      HTPO2(INFLOW)=HTPO2X
      QCENT(INFLOW)=QCENTX
      TPR(INFLOW)=TPRX
      JN=INFLOW
70    NWTYP(N)=10
      DO 75 K=1,NB
        IF (NO(K).EQ.NBR.AND.LA(K).GT.30.0)
     .    WRITE (8,220) NO(K),LA(K),TS
```

```
75      CONTINUE
        WRITE (8,160) NBR,TS,TPR(JN),CONT(JN),CONC(JN),HEAT(JN),
     .                  O2MIN(JN),SMPO2(JN),HTPO2(JN),QCENT(JN)
C
      ELSE IF (ICODE.EQ.4) THEN
        IF (INFLOW.GT.0) THEN
          DO 90 I=1,INFLOW
            IF (NCENT(I).EQ.NBR) THEN
              DO 80 J=I,INFLOW
                NCENT(J)=NCENT(J+1)
                CONT(J)=CONT(J+1)
                CONC(J)=CONC(J+1)
                HEAT(J)=HEAT(J+1)
                O2MIN(J)=O2MIN(J+1)
                SMPO2(J)=SMPO2(J+1)
                HTPO2(J)=HTPO2(J+1)
                QCENT(J)=QCENT(J+1)
                TPR(J)=TPR(J+1)
80            CONTINUE
              INFLOW=INFLOW-1
              NWTYP(N)=0
              WRITE (8,170) NBR,TS
              GO TO 100
            ENDIF
90        CONTINUE
        ENDIF
      ELSE
        WRITE (8,180) ICODE, NBR
      ENDIF
100   IREAD=1
      GO TO 1
C
C
110   FORMAT (/,1X,'* ATTENTION *  AIRWAY',I5,' BECAME AN ORDINARY ',
     .        'AIRWAY WITH',/,1X,'R = ',F8.3,' AT',F6.1,' MIN. AFTER ',
     .        'EVENT.')
120   FORMAT (/,1X,'* WARNING *  INVALID NUMBER OF DATA POINTS FOR A ',
     .        'FAN CURVE,',/,1X,'WHICH IN TIME TABLE IS 1 TO 5.')
130   FORMAT (/,1X,'* WARNING *  NUMBER OF FANS BECAME EXCESSIVE. ',
     .        'FAN IN',/,1X,'BRANCH ',I5,' IS TAKEN AS FIXED-P FAN OF',
     .        F7.3,' IN.W.G.')
140   FORMAT (/,1X,'* ATTENTION * THE FOLLOWING CHARACTERISTICS FOR ',
     .        'FAN ',I4,/,' ACTIVATED AT',F6.2,' MIN. AFTER EVENT.',//,
     .        5(' Q/1000  PF '),//,5(F8.1,F6.2))
150   FORMAT (/,1X,'* ATTENTION *  AIRWAY',I5,' BECAME A FAN BRANCH ',
     .        'WITH PRESSURE',/,' FIXED AT',F6.2,' IN. W.G. AT',F6.2,
     .        ' MIN. AFTER EVENT.')
160   FORMAT (/,1X,'* ATTENTION *  BRANCH',I5,' CAUGHT IN FIRE AT ',
     .        F6.2,' MIN.',/,1X,'LEADING TIME: ',F5.1,'  PARAMETERS AS',
     .        ' FOLLOWS:',//,T7,'FIXED-HEAT-INPUT',T30,'OXYGEN-RICH',
     .        T48,'FUEL-RICH',/,1X,'FLOWRATE',T14,'CONC.'
     .        ,T23,'HEAT',T34,'(%)',T43,'FUME P O2',T54,'HEAT PO2',T66,
     .        'REF. Q',//,1X,F8.0,T11,F8.2,T21,E8.3,T30,F7.2,T43,
     .        F7.2,T53,F7.2,T63,F9.0)
```

```
170   FORMAT (/,1X,'* ATTENTION *  FIRE',I5,' CEASED AT',F6.2,' MIN.',
     .        ' AFTER EVENT.')
180   FORMAT (/,' * ATTENTION *  INVALID CODE',I5,' FOR AIRWAY',I5,
     .        ' IGNORED. (CDCH)')
190   FORMAT (/,1X,'* ATTENTION *  SIMULATION STEP WAS CHANGED ',
     .        'FROM ',F6.2,' SEC.',/,' TO ',F6.2,' SEC. AT ',F6.2,
     .        ' MINS. AFTER EVENT.')
200   FORMAT (/,1X,'* WARNING *  INVALID SIMULATION STEP (',F6.2,' SEC',
     .        ') IGNORED.',/,1X,'IT MUST BE LESS THAN 30.0 SEC. (CDCH)')
210   FORMAT (/,1X,'* ATTENTION *  AIRWAY',I5,' NOT FOUND, DATA ',
     .        'IGNORED. (CDCH)')
220   FORMAT (/,1X,'* ATTENTION *  AIRWAY',I5,' OF LENGTH',F8.0,
     .        ' FT CONTAINS A',/,' HEAT/CONTAMINANT SOURCE AT ',F6.2,
     .        ' MIN. AFTER EVENT.',/,' IT IS ADVISED TO DIVIDE THE ',
     .        'AIRWAY INTO TWO AND LOCATE THE',/,' FIRE SOURCE IN ',
     .        'THE ONE SHORTER THAN 30 FT.')
230   FORMAT (/,1X,'* ATTENTION *  FIRE BRANCH ',I5,' NOT FOUND  AT ',
     .        F6.2,' MIN.',/,' AFTER SIMULATION, CHANGE IGNORED.')
240   FORMAT (/,1X,'* ATTENTION *  TOO MANY JDPP INPUT, DATA IGNORED.')
250   FORMAT (/,1X,'* ATTENTION *  JUNCTION',I6,' NOT FOUND AT',F6.2,
     .        ' MIN.',/,' AFTER EVENT, CHANGE IGNORED.')
260   FORMAT (/,1X,'* WARNING *  NUMBER OF FIRE SOURCES BECAME ',
     .        'EXCESSIVE. SPECIFIED',/,' CONDITION CHANGE IN TIME ',
     .        'TABLE IGNORED.')
300   FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ',
     .        ' IN:',//)
310   FORMAT (1X,'TS,1,NBR,R')
320   FORMAT (1X,'TS,2,NBR,INDEX,QF,PF,QF,PF,QF,PF,QF,PF,QF,PF')
330   FORMAT (1X,'TS,3,NBR,CONT,CONC,HEAT,02MIN,SMPO2,HTPO2,QCENT')
340   FORMAT (1X,'TS,4,NBR')
350   FORMAT (1X,'TS,5,DELTAT')
360   FORMAT (1X,'TS,6,TOUT')
365   FORMAT (1X,'TS,7,JDPP')
370   FORMAT (/,1X,'THE CURRENT INPUT WAS:',//,80A1)
C
      END
C
C
      SUBROUTINE  DTTR (MARKL,MARKP)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       REARRANGING DETAILED DATA RECORDS FOR THE AIRWAY WHOSE AIRFLOW
C     REVERSED.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
      DO 90 I=1,NB
         DO 1 J=1,NJ
            IF (JNO(J).EQ.JS(I)) JU=J
            IF (JNO(J).EQ.JF(I)) JW=J
```

```
1       CONTINUE
C
C   MARKP=0: A THERMALLY BALANCED SOLUTION HAS NOT BEEN REACHED YET
C            WITHIN A TIME INTERVAL.
C   MARKP=1: THE CALCULATION OF THE PRESENT TIME INTERVAL IS DONE.
C   Q(I): AIRFLOW RATE IN AIRWAY NO(I) AT THE END OF THE PRESENT
C         INTERVAL.
C   QQ(I): TIME AVERAGED AIRFLOW RATE.
C
        IF (MARKP.EQ.0) QRS=QQ(I)
        IF (MARKP.EQ.1) QRS=Q(I)
        IF (QRS.LT.0.0) THEN
          MARKL=1
          IQ=IBTN(I)
          IF (IQ.GT.0) THEN
            IF (MARKP.EQ.1) THEN
              IS=NSAC(I,2)
              IF (IS.GT.0) THEN
                DO 2 J=1,IS
                  DO 2 K=1,5
                    GBTN(J,K)=DPPA(IQ,J,K)
2               CONTINUE
                DO 6 J=1,IS
                  JP=IS-J+1
                  DO 4 K=1,5
                    DPPA(IQ,JP,K)=GBTN(J,K)
4                 CONTINUE
                  DPPA(IQ,JP,1)=LA(I)-DPPA(IQ,JP,1)
                  IF (J.EQ.1) THEN
                    DPPA(IQ,JP,2)=TAUXC(IQ)
                  ELSE
                    DPPA(IQ,JP,2)=TAUXA(IQ,J-1)
                  ENDIF
6               CONTINUE
                DO 7 J=1,IS
                  JP=IS-J+1
                  IF (J.EQ.1) THEN
                    TAUXA(IQ,JP)=TRD(I)
                  ELSE
                    TAUXA(IQ,JP)=GBTN(J-1,2)
                    IF (J.EQ.IS) TAUXC(IQ)=GBTN(J,2)
                  ENDIF
7               CONTINUE
                FMASS1=DPPA(IQ,1,5)
                IF (IS.GE.2) THEN
                  DO 8 J=1,IS-1
                    DPPA(IQ,J,5)=DPPA(IQ,J+1,5)
8                 CONTINUE
                ENDIF
                IF (FMASS(IQ).LT.0.0) THEN
                  FK1=535.0/(460.0+(DPPA(IQ,IS,2)+TRD(I))/2.0)
                  FMASS(IQ)=A(I)*DPPA(IQ,IS,1)*0.075*FK1
                ENDIF
                DPPA(IQ,IS,5)=FMASS(IQ)
```

```
            FMASS(IQ)=FMASS1
            RDPROP(I)=DPPA(IQ,1,3)
            RDCH4(I)=DPPA(IQ,1,4)
            RDOP(I)=DPPA(IQ,1,3)
            RCH4(I)=DPPA(IQ,1,4)
         ELSE
            IF (NWTYP(I).NE.10) THEN
               RDPROP(I)=PROP(JU)
               RDCH4(I)=PRCH4(JU)
               RDOP(I)=PROP(JU)
               RCH4(I)=PRCH4(JU)
            ENDIF
         ENDIF
      ELSE
         IS=NSACB(I,2)
         IF (IS.GT.0) THEN
            DO 10 J=1,IS
               DO 10 K=1,5
                  GBTN(J,K)=DPPB(IQ,J,K)
10          CONTINUE
            DO 30 J=1,IS
               JP=IS-J+1
               DO 20 K=1,5
                  DPPB(IQ,JP,K)=GBTN(J,K)
20          CONTINUE
            DPPB(IQ,JP,1)=LA(I)-DPPB(IQ,JP,1)
            IF (J.EQ.1) THEN
               DPPB(IQ,JP,2)=TAUXD(IQ)
            ELSE
               DPPB(IQ,JP,2)=TAUXB(IQ,J-1)
            ENDIF
30          CONTINUE
            DO 32 J=1,IS
               JP=IS-J+1
               IF (J.EQ.1) THEN
                  TAUXB(IQ,JP)=TTRD(I)
               ELSE
                  TAUXB(IQ,JP)=GBTN(J-1,2)
                  IF (J.EQ.IS) TAUXD(IQ)=GBTN(J,2)
               ENDIF
32          CONTINUE
            FMASS1=DPPB(IQ,1,5)
            IF (IS.GE.2) THEN
               DO 35 J=1,IS-1
                  DPPB(IQ,J,5)=DPPB(IQ,J+1,5)
35             CONTINUE
            ENDIF
            IF (FMASS(IQ).LT.0.0) THEN
               FK1=535.0/(460.0+(DPPB(IQ,IS,2)+TTRD(I))/2.0)
               FMASS(IQ)=A(I)*DPPB(IQ,IS,1)*0.075*FK1
            ENDIF
            DPPB(IQ,IS,5)=FMASS(IQ)
            FMASS(IQ)=FMASS1
            DO 40 J=1,IS
```

```
                        DPPA(IQ,J,5)=DPPB(IQ,J,5)
40                    CONTINUE
                      RDPROP(I)=DPPB(IQ,1,3)
                      RDCH4(I)=DPPB(IQ,1,4)
                      RDOP(I)=DPPB(IQ,1,3)
                      RCH4(I)=DPPB(IQ,1,4)
                    ELSE
                      IF (NWTYP(I).NE.10) THEN
                        RDPROP(I)=PROPS(JU)
                        RDCH4(I)=PRCH4S(JU)
                        RDOP(I)=PROPS(JU)
                        RCH4(I)=PRCH4S(JU)
                      ENDIF
                    ENDIF
                  ENDIF
                ENDIF
                TRDA=TRD(I)
                TTRDA=TTRD(I)
                TRD(I)=TJS(I)
                TTRD(I)=TTJS(I)
                TJS(I)=TRDA
                TTJS(I)=TTRDA
                JE=JS(I)
                JS(I)=JF(I)
                JF(I)=JE
                DZRD(I)=-DZRD(I)
                FRNVP(I)=-FRNVP(I)
                FFRNVP(I)=-FFRNVP(I)
                QTP(I)=-QTP(I)
                QQ(I)=-QQ(I)
                Q(I)=-Q(I)
C
C     NREV(I): NUMBER OF CHANGES IN AIRFLOW DIRECTION (QQ) IN AIRWAY
C              NO(I) IN THE PRESENT TIME INTERVAL.
C     NNREV(I): NUMBER OF CHANGES IN AIRFLOW DIRECTION (Q) IN AIRWAY
C                NO(I) SINCE TIME ZERO.
C
              NREV(I)=NREV(I)+1
            ENDIF
            IF (MARKP.EQ.1.AND.Q(I)*QTP(I).LT.0.0) NNREV(I)=NNREV(I)+1
            IF (ABS(QQ(I)).LT.1.E-5) THEN
              WRITE (8,200) NO(I),QQ(I)
              QQ(I)=1.E-5
            ENDIF
            IF (ABS(Q(I)).LT.1.E-5) THEN
              WRITE (8,200) NO(I),Q(I)
              Q(I)=1.E-5
            ENDIF
90      CONTINUE
C
        IF (MARKL.NE.0) THEN
          L=0
          M=0
          N=1
```

```
      DO 120 I=1,MAXJ
         K=L
         DO 100 J=1,NB
            IF (JS(J).EQ.I) THEN
               L=L+1
               NGOUT(L)=J
            ENDIF
100      CONTINUE
         LOUT(N)=L
         MM=M
         DO 110 J=1,NB
            IF (JF(J).EQ.I) THEN
               M=M+1
               NGIN(M)=J
            ENDIF
110      CONTINUE
         MMIN(N)=M
         IF (MM.NE.M.OR.K.NE.L) THEN
           JNOL(N)=I
           N=N+1
         ENDIF
120   CONTINUE
      DO 140 I=1,NJ
         DO 130 J=1,NJ
            IF (JNOL(I).EQ.JNO(J)) THEN
               JLR(I)=J
               GO TO 140
            ENDIF
130      CONTINUE
140   CONTINUE
   ENDIF
C
200 FORMAT (//,1X,'* ATTENTION *  AIRWAY',I5,' HAD ITS Q OR QQ',
   .          ' DOWN TO',F6.3,/,' FT3/MIN. DURING ITERATIONS. IT WAS ',
   .          'SET TO 1E-5 FT3/MIN.',/,' TO AVOID POSSIBLE DIVISION ',
   .          'FAULT DURING ITERATION.')
C
   RETURN
   END
C
C
   SUBROUTINE  CDJN (ISQC,MARKS)
C
C  ------------------------------------------------------------------------------------------------------------------------
C
C     SUBROUTINE PURPOSES:
C     1) EVALUATION OF THERMAL STATES IN JUNCTIONS.
C     2) DEVELOPING A NEW DATA RECORD INTO AIRWAYS (A) WHOSE BEGINNING
C        JUNCTION HAS DRASTIC CONDITION CHANGES; OR (B) WHERE AIRFLOW
C        REVERSAL HAPPENED; OR (C) WHERE A FIRE SOURCE EXISTS; OR (D)
C        WHICH GOT DETAILED RECORDS IN PREVIOUS INTERVALS ALREADY.
C
C  ------------------------------------------------------------------------------------------------------------------------
C
```

```
$INCLUDE: 'CMMN.DAT'
C
      CP=0.24
      DO 60 I=1,NJ
         K=JLR(I)
         JY=1
         IF (I.GT.1) JY=LOUT(I-1)+1
         JZ=LOUT(I)
         DO 50 J=JY,JZ
            NW=NGOUT(J)
            IF (NW.LE.O) WRITE (8,*) 'NW',I,JY,JZ,NGOUT(J),J
            CO2=0.0
            WR=0.0
C     NWTYP(I)=10 INDICATES THAT AIRWAY NO(I) CONTAINS A FIRE SOURCE.
C     NSAC(I,2): NUMBER OF CONTROL VOLUMES IN AIRWAY NO(I).
C     IBTN(I) STORES THE ADDRESS OF DATA RECORDS FOR AIRWAY NO(I) IN
C               ARRAY DPPA, DPPB ETC.
C
            IF (A(NW).GE.1000.0) GO TO 50
            IF (MARKS.NE.O.AND.NWTYP(NW).NE.10) THEN
               IF (JCH(K).EQ.O) GO TO 50
C
C       WHEN THE NEWLY OBTAINED DATA NEED TO BE UPDATED AGAIN IN THE
C     PRESENT TIME INTERVAL, THE DATA OBTAINED IN THE PREVIOUS SEARCH
C     IN THE INTERVAL ARE ABANDONED TO AVOID DUPLICATED DATA
C     RECORDS.
C
               IF (IBTN(NW).GT.O) THEN
                  IZ=IBTN(NW)
                  NSAC(NW,2)=NSACB(NW,2)
                  DO 5 IX=1,10
                     DO 5 IY=1,5
                        DPPA(IZ,IX,IY)=DPPB(IZ,IX,IY)
5                 CONTINUE
               ENDIF
            ELSE IF (NWTYP(NW).NE.10) THEN
C
C     NON-QUALIFIED AIRWAYS GET NO DETAILED DATA RECORDS AND ARE IGNORED
C
               IF (JDP.GT.O) THEN
                  DO 7 IX=1,JDP
                     IF (NO(NW).EQ.JDPP(IX)) GO TO 8
7                 CONTINUE
               ENDIF
               IF (JCH(K).EQ.O.AND.IBTN(NW).EQ.O) GO TO 50
            ENDIF
8           CONTAM=0.0
            CONTQ=0.0
            HEATAD=0.0
            TFS=0.0
C
C  ** CHECK TO SEE IF A FIRE SOURCE EXISTS IN THE AIRWAYS.
C
```

```
      DO 10 L=1,INFLOW
         IF (NCENT(L).EQ.NO(NW)) GO TO 20
10    CONTINUE
      GO TO 30
C
20    NM=1
      FACT=1.0
      IF (QCENT(L).GT.10.0) THEN
         Q1=QQ(NW)
         IF (Q1.LT.QCENT(L)) Q1=Q1+QCENT(L)
         FACT=1.0+(QQ(NW)-QCENT(L))/Q1
      ENDIF
      IF (ABS(CONT(L)).GT.1.E-3) THEN
         CONTAM=CONT(L)*CONC(L)/100.0
         CONTQ=CONT(L)
      ELSE
         CONTAM=0.0
      ENDIF
      O2BEH(L)=(0.21-PROP(K)-CONTAM/QQ(NW))*100.0
      HEATAD=HEAT(L)*FACT
C
      IF (O2MIN(L).GT.1.0) THEN
         O2SCL=O2MIN(L)+(21.0-O2MIN(L))*(FACT-1.0)
         TACCA=TACC+DELTAT
         IF (TACCA.GT.TPR(L)*60.0) TACCA=TPR(L)*60.0
         PROPA=PROP(K)
         O2CONS=((0.21-PROPA-O2SCL/100.0)*TACCA/
                 (TPR(L)*60.0))*QQ(NW)
         IF (O2CONS.LT.0.0) O2CONS=0.0
         O2BEH(L)=(0.21-PROPA-O2CONS/QQ(NW))*100.0
         WR=0.0
         HEATAD=O2CONS*437.0
         CONTAM=PROPA*QQ(NW)+O2CONS
      ENDIF
C
      IF (SMPO2(L).GT.1.E-3) THEN
         O2LT=0.21-0.21*(TACC+DELTAT)/(TPR(L)*60.0)
         IF (O2LT.LT.0.0) O2LT=0.0
         PROPA=PROP(K)
         O2CONS=(0.21-O2LT-PROPA)*QQ(NW)
         IF (O2CONS.LT.0.0) O2CONS=0.0
         CONTAM=(0.21-O2LT)*QQ(NW)
         IF ((0.21-O2LT).LT.PROPA) CONTAM=PROPA*QQ(NW)
         O2BEH(L)=O2LT*100.0
         IF (FACT.LE.1.0) THEN
            HTSCL=HTPO2(L)
         ELSE
            HTSCL=HTPO2(L)/FACT
         ENDIF
         WR=0.0
         HEATAD=O2CONS*HTSCL
      ENDIF
C
30    TK=T(K)
```

```
            IF (NWTYP(NW).EQ.10) TK=TAS(K)
            VART=(9900.0+TK)**2+2.0*HEATAD/(QQ(NW)*2.4E-5*DR)
            TJS(NW)=-9900.0+SQRT(VART)
            IF (CONCT.EQ.1) TJS(NW)=TJSS(NW)
            IF (NWTYP(NW).EQ.10) THEN
               TRD(NW)=TJS(NW)
               RDPROP(NW)=(PROP(K)*QQ(NW)+CONTAM)/(QQ(NW)+CONTQ)
               RDCH4(NW)=PRCH4(K)
               HTAD(L)=(TJS(NW)-T(K))*0.24*QQ(NW)*DR
               GO TO 50
            ENDIF
C
C
            NTAUXC=0
            IF (NSAC(NW,1).EQ.0) THEN
C
C     NSAC(I,1)=1 INDICATES THAT THE FUME FRONT IS WITHIN AIRWAY NO(I).
C
               NTAUXC=1
               NSAC(NW,1)=1
            ENDIF
            NSAC(NW,2)=NSAC(NW,2)+1
            NS=NSAC(NW,2)
            IF (IBTN(NW).EQ.0) THEN
               ISQC=ISQC+1
               IF (ISQC.LT.NMZ) THEN
                  IBTN(NW)=ISQC
               ELSE
C
C        WHEN THE CAPACITIES OF DATA RECORDING ARRAYS ARE SATURATED, LATE
C     REQUEST WILL BE TURNED DOWN AND A MESSAGE WILL BE GIVEN.
C
                  NSAC(NW,1)=0
                  NSAC(NW,2)=0
                  IF (IAC.GT.0) THEN
                     DO 35 I1=1,IAC
                        IF (IOMIT(I1).EQ.NW) GO TO 36
35                   CONTINUE
                  ENDIF
                  IF (IAC.GE.IMX) GO TO 50
                  IAC=IAC+1
                  I1=IAC
36                IOMIT(I1)=NW
                  ROMIT(I1,1)=T(K)
                  ROMIT(I1,2)=PROP(K)
                  ROMIT(I1,3)=PRCH4(K)
                  GO TO 50
               ENDIF
            ENDIF
            NQ=IBTN(NW)
            NN=0
C
C
            TS=DELTAT/60.0
```

```
            IF (TLEFT(K).GE.0.0) TS=TLEFT(K)+0.00001
            DPPA(NQ,NS,5)=DR*QQ(NW)*TS
            IF (NS.GT.1.AND.DELTAT/60.0.GT.TS) THEN
               DPPA(NQ,NS-1,5)=DPPB(NQ,NS-1,5)+DR*QQ(NW)*
     .                         (DELTAT/60.0-TS)
            ENDIF
            TAUXA(NQ,NS)=TJS(NW)
            CALL COLUMN (TJS(NW),SEG,NW,NQ,NS)
            FE=SEG/LA(NW)
            SEGZ=QQ(NW)*TS*(TMRD(NW)+460.0)/(A(NW)*(TR+460.0))
            FA=XNEW(NW)*SEGZ/CP
            IF (FA.LT.25.0) THEN
               FAS=EXP(-FA)
            ELSE
               FAS=0.0
            ENDIF
C
C  AIR TEMP. OF THE C.V. AT ITS NEWLY ARRIVED POSITION IS ESTIMATED.
C
            DPPA(NQ,NS,2)=TROCK(NW)+(TJS(NW)-TROCK(NW))*FAS-
     .                    (DZRD(NW)*FE/(2.0*778.26*CP))*(1.0+FAS)
C
C     DATA RECORDED IN DPPA: (1) LOCATION, (2) TEMP. (3) CH4,
C                            (4) FUME, (5) AIR MASS IN C.V..
C
            IF (NTAUXC.EQ.1) TAUXC(NQ)=TROCK(NW)+(TTJS(NW)-TROCK(NW))*
     .                       FAS-(DZRD(NW)*FE/(2.0*778.26*CP))*(1.+FAS)
            DPPA(NQ,NS,1)=SEG
            DPPA(NQ,NS,3)=(PROP(K)*QQ(NW)+CONTAM)/(QQ(NW)+CONTQ)
            DPPA(NQ,NS,4)=(PRCH4(K)*QQ(NW)+CH4V(NW)*FE)/
     .                    (QQ(NW)+CH4V(NW)*FE)
50          CONTINUE
60       CONTINUE
C
         RETURN
         END
C
C
         SUBROUTINE COLUMN (T1,X3,I,JQ,JP)
C
C        ------------------------------------------------------------------------------------------------------------------------
C
C         SUBROUTINE PURPOSE:
C           DETERMINING THE ADVANCING DISTANCE OF AIR COLUMNS.
C
C        ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
         IF (DPPA(JQ,JP,5).LT.3.0) THEN
            X3=0.1
            RETURN
         ENDIF
         KK=0
```

```
      X1=1.0
      X2=100.0
      CP=0.2376+0.000024*TMRD(I)
      C1=TROCK(I)+460.0
      C2=T1-TROCK(I)
      C3=-HK(I)*DCOAGE(I)*O(I)**2/(DR*QQ(I)*CP*120.0*A(I))
      AC=DPPA(JQ,JP,5)*53.34/(144.0*14.7*A(I))
10    EXPR=0.0
      IF (C3*X1.GT.-25.0) EXPR=EXP(C3*X1)
      XXX=(C1+C2*EXPR)/(C1+C2)
      ALOGR=0.0
      IF (ABS(XXX-1.0).GT.1.E-4) ALOGR=ALOG(XXX)
      QM1=X1/C1-ALOGR/(C1*C3)
      IF (QM1.GT.AC) THEN
         X1=X1/5.0
         IF (X1.GT.1.E-3) GO TO 10
      ENDIF
20    EXPR=0.0
      IF (C3*X2.GT.-25.0) EXPR=EXP(C3*X2)
      XXX=(C1+C2*EXPR)/(C1+C2)
      ALOGR=0.0
      IF (ABS(XXX-1.0).GT.1.E-4) ALOGR=ALOG(XXX)
      QM2=X2/C1-ALOGR/(C1*C3)
      IF (QM2.LT.AC) THEN
         X2=X2*2.0
         IF (X2.LT.10000.0) GO TO 20
      ENDIF
25    X4=X2/5.0
      EXPR=0.0
      IF (C3*X4.GT.-25.0) EXPR=EXP(C3*X4)
      XXX=(C1+C2*EXPR)/(C1+C2)
      ALOGR=0.0
      IF(ABS(XXX-1.0).GT.1.E-4) ALOGR=ALOG(XXX)
      QM4=X4/C1-ALOGR/(C1*C3)
      IF (QM4.GT.AC) THEN
         X2=X4
         QM2=QM4
         IF (X2.GT.10.0) GO TO 25
      ENDIF
30    X3=X1+(X2-X1)*(AC-QM1)/(QM2-QM1)
      EXPR=0.0
      IF (C3*X3.GT.-25.0) EXPR=EXP(C3*X3)
      XXX=(C1+C2*EXPR)/(C1+C2)
      ALOGR=0.0
      IF (ABS(XXX-1.0).GT.1.E-4) ALOGR=ALOG(XXX)
      QM3=X3/C1-ALOGR/(C1*C3)
      IF (ABS(QM3-AC).GT.(5.E-4*AC)) THEN
         IF (QM3.LE.AC) THEN
            X1=X3
            QM1=QM3
         ELSE
            X2=X3
            QM2=QM3
         ENDIF
```

```
        KK=KK+1
        IF (KK.LE.40) GO TO 30
      ENDIF
      IF (X3.LT.0.1) X3=0.1
C
C
      RETURN
      END
C
C
      SUBROUTINE  ADPT (IS,IQ,MARKS,I)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSES:
C        1) ADVANCING EXISTING C.V. IN AIRWAYS, UPDATING RELEVANT DATA.
C        2) UPDATING AIRWAY ENDING CONDITIONS.
C        3) CONDENSING DATA STORAGE ARRAYS IF THERE IS A C.V. EXCEEDING
C           THE AIRWAY ENDING.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
C     IS: NUMBER OF C.V. IN AIRWAY NO(I).
C
      CP=0.24
      IF (NSAC(I,2).EQ.0) RETURN
      DO 5 K=1,NJ
          IF (JS(I).EQ.JNO(K)) GO TO 8
5     CONTINUE
8     IF (MARKS.GT.0.AND.JCH(K).EQ.0) RETURN
      IF (IS.LE.0) GO TO 50
      IF (DPPA(IQ,IS+1,1).GT.LA(I)) GO TO 50
C
C       "MARKS" ACTS AS A TIMER. WHEN MARKS=0, THIS SUB. ADVANCES ALL
C     EXISTING C.V., WHILE MARKS>0, IT SEARCHES FOR THOSE C.V. WHICH
C     WERE GENERATED IN THE PRESENT INTERVAL AND TRAVELLED MORE THAN
C     ONE AIRWAY IN THE PRESENT INTERVAL THEN UPDATES THE RELEVANT DATA.
C
C ** MARKS=0, ADVANCING ALL EXISTING DATA POINTS, STARTING WITH THE C.V
C     CLOSEST TO THE BEGINNING JUNCTION OF THE AIRWAY.
C
      SEGZ=QQ(I)*DELTAT*(TMRD(I)+460.0)/(60.0*A(I)*(TR+460.0))
      FA=XNEW(I)*ABS(SEGZ)/CP
      FAS=0.0
      IF (FA.LT.25.0) FAS=EXP(-FA)
C
      DO 40 K=1,IS
          JA=IS-K+1
          NN=0
          MK=0
          SEG=DPPA(IQ,JA+1,1)-DPPB(IQ,JA+1,1)
          IF (ABS(SEG).LT.1.E-3) SEG=0.001
```

```
      VL=SEG/LA(I)
      T0=TAUXB(IQ,JA)
      T2=DPPB(IQ,JA,2)
      IRV=0
      IF ((TAUXB(IQ,JA).LT.DPPB(IQ,JA,2).AND.TAUXB(IQ,JA).GT.
     .   TROCK(I)).OR.(TAUXB(IQ,JA).GT.DPPB(IQ,JA,2).AND.
     .   TAUXB(IQ,JA).LT.TROCK(I))) THEN
         T0=DPPB(IQ,JA,2)
         T2=TAUXB(IQ,JA)
         IRV=1
      ENDIF
      T1=TROCK(I)+(T0-TROCK(I))*FAS-(DZRD(I)*VL/(2*778.26*CP))
     .    *(1.0+FAS)
      TAUXA(IQ,JA)=T1
      CALL COLUMN (T1,SEG1,I,IQ,JA)
      SEG=DPPA(IQ,JA+1,1)+SEG1-DPPB(IQ,JA,1)
      VL=SEG/LA(I)
      DPPA(IQ,JA,2)=TROCK(I)+(T2-TROCK(I))*FAS-
     .                (DZRD(I)*VL/(2*778.26*CP))*(1.0+FAS)
      IF (JA.EQ.1) TAUXC(IQ)=TROCK(I)+(TAUXD(IQ)-TROCK(I))*FAS-
     .                (DZRD(I)*VL/(2*778.26*CP))*(1.0+FAS)
      IF (IRV.EQ.1) THEN
         TBW=DPPA(IQ,JA,2)
         DPPA(IQ,JA,2)=TAUXA(IQ,JA)
         TAUXA(IQ,JA)=TBW
      ENDIF
      DPPA(IQ,JA,1)=DPPB(IQ,JA,1)+SEG
      SE=ABS(DPPA(IQ,JA,1)-DPPB(IQ,JA,1))
      DPPA(IQ,JA,4)=(DPPB(IQ,JA,4)*QQ(I)+CH4V(I)*SE/LA(I))/
     .                (QQ(I)+CH4V(I)*SE/LA(I))
      IF (DPPA(IQ,JA,1).GE.LA(I)) GO TO 50
40    CONTINUE
      VL=(DPPA(IQ,1,1)-DPPB(IQ,1,1))/LA(I)
      TAUXC(IQ)=TROCK(I)+(TAUXD(IQ)-TROCK(I))*FAS-(DZRD(I)*VL/
     .          (2.0*778.26*CP))*(1.0+FAS)
50    ISS=NSAC(I,2)
      DO 52 K=1,ISS
         JA=ISS-K+1
         IF (DPPA(IQ,JA,1).GE.LA(I)) GO TO 54
52    CONTINUE
      IF (NSAC(I,1).EQ.1) RETURN
C
C     IF NO C.V. EXCEEDS THE ENDING, THEN SUBROUTINE RETURNS AFTER
C     UPDATING AIRWAY ENDING CONDITIONS.
C
      SEG=LA(I)-DPPA(IQ,1,1)
      SSL=LA(I)-DPPB(IQ,1,1)
      IF (SSL.LE.SEG) SSL=SEG
      FA=XNEW(I)*SEG/CP
      IF (FA.LE.25.0) THEN
         FAS=EXP(-FA)
      ELSE
         FAS=0.0
      ENDIF
```

```
      TT=DPPA(IQ,1,2)
      ENDT=TROCK(I)+(TT-TROCK(I))*FAS-(DZRD(I)*SEG/
     .          (2*778.26*CP*LA(I)))*(1.0+FAS)
      TRD(I)=TTRD(I)+(ENDT-TTRD(I))*(SSL-SEG)/SSL
      IF (CONCT.EQ.1) TRD(I)=TRDD(I)
      RDPROP(I)=RDOP(I)+(DPPA(IQ,1,3)-RDOP(I))*(SSL-SEG)/SSL
      RETURN
C
C ** C.V. EXCEEDS THE ENDING, UPDATE OF AIRWAY ENDING CONDITIONS
C     IS CALLED FOR.
C     TLEFT(J): TIME LEFT AT JUNCTION JNO(J) FOR WAVE FRONT (C.V.)
C                 TO TRAVEL INTO THE DOWN-STREAM AIRWAY.
C
54    DO 56 LL=1,NJ
          IF (IABS(JF(I)).EQ.JNO(LL)) KL=LL
          IF (IABS(JS(I)).EQ.JNO(LL)) ML=LL
56    CONTINUE
      FL=DPPA(IQ,JA,1)-LA(I)
      TK=DELTAT*FL/((DPPA(IQ,JA,1)-DPPB(IQ,JA,1))*60.0)
      IF (TK.GT.TLEFT(KL)) TLEFT(KL)=TK
      IF (NSAC(I,1).NE.(-1)) NSAC(I,1)=-1
C
C     NSAC(I,1)=-1: AIRWAY NO(I) IS A FUME-FILLED ONE.
C     RDPROP(I): FUME CONCENTRATION AT AIRWAY ENDING.
C     RDCH4(I): CH4 CONCENTRATION AT AIRWAY ENDING.
C     DPPA,DPPB: (1) DISTANCE, (2) TEMP. (3) FUME, (4) CH4,
C                   (5) AIR MASS IN C.V.
C
      RDCH4A=DPPA(IQ,JA,4)-CH4V(I)*(DPPA(IQ,JA,1)-LA(I))/
     .          (LA(I)*QQ(I))
C
C     NSACB: BACKUP ARRAY OF NSAC. NSACB(I,2) INDICATES THE NUMBER
C             OF C.V. IN AIRWAY NO(I) IN THE LAST INTERVAL.
C
      IF (JA.EQ.NSAC(I,2)) THEN
         SECT=LA(I)
         TLAST=TJS(I)
         RDF=DPPA(IQ,JA,3)-PROP(ML)
         IF (NWTYP(I).EQ.10) RDF=0.0
         SSL=DPPA(IQ,JA,1)
      ELSE
         SECT=LA(I)-DPPA(IQ,JA+1,1)
         TLAST=TAUXA(IQ,JA)
         RDF=DPPA(IQ,JA,3)-DPPA(IQ,JA+1,3)
         SSL=DPPA(IQ,JA,1)-DPPA(IQ,JA+1,1)
      ENDIF
      RDPROP(I)=DPPA(IQ,JA,3)-RDF*FL/SSL
      RDCH4(I)=RDCH4A
C
      IF ((TAUXB(IQ,JA).LT.DPPB(IQ,JA,2).AND.TAUXB(IQ,JA).GT.
     .   TROCK(I)).OR.(TAUXB(IQ,JA).GT.DPPB(IQ,JA,2).AND.
     .   TAUXB(IQ,JA).LT.TROCK(I))) THEN
C
         FL1=DPPB(IQ,JA,1)-DPPB(IQ,JA+1,1)
```

```
      FLL=(DPPA(IQ,JA,1)-LA(I))/(DPPA(IQ,JA,1)-DPPA(IQ,JA+1,1))
      TLAST=DPPB(IQ,JA,2)+(TAUXB(IQ,JA)-DPPB(IQ,JA,2))*FLL
      SECT=LA(I)-DPPB(IQ,JA,1)+FL1*FLL
    ENDIF
    FA=XNEW(I)*SECT/CP
    FAS=0.0
    IF (FA.LT.25.0) FAS=EXP(-FA)
    VL=SECT/LA(I)
    TRD(I)=TROCK(I)+(TLAST-TROCK(I))*FAS-
   .        (DZRD(I)*VL/(2*778.26*CP))*(1.0+FAS)
    IF (CONCT.EQ.1) TRD(I)=TRDD(I)
C
C      WHEN A C.V EXCEEDS AIRWAY ENDING, THE DATA RELATED TO THAT C.V.
C   ARE DISCARDED AND DATA RECORDING ARRAYS ARE CONDENSED.
C
    JAA=JA+1
    DO 80 K=JAA,11
      IF (K.EQ.JAA) TAUXC(IQ)=TAUXA(IQ,K-1)
      IF (K.LE.10) THEN
        KA=K-JAA+1
        TAUXA(IQ,KA)=TAUXA(IQ,K)
        DO 60 L=1,5
          DPPA(IQ,KA,L)=DPPA(IQ,K,L)
60      CONTINUE
      ELSE
        TAUXA(IQ,10)=0.0
        DO 70 L=1,5
          DPPA(IQ,10,L)=0.0
70      CONTINUE
      ENDIF
80  CONTINUE
    NSAC(I,2)=NSAC(I,2)-JA
C
    RETURN
    END
C
C
    SUBROUTINE  PARAM
C
C   ------------------------------------------------------------------------------------------------------------------------
C
C      SUBROUTINE PURPOSES:
C      1) EVALUATION OF MEAN AND MEAN-SQUARE TEMP. IN AIRWAYS.
C      2) CALCULATION OF VARIED AIRFLOW RESISTANCE AND THE PRODUCT
C         OF MEAN TEMP. AND ELEVATION VARIATION OF EACH AIRWAY FOR
C         ASSESSING NVP.
C
C   ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
    CP=0.24
    DO 30 I=1,NB
      IF (NWTYP(I).EQ.10) THEN
```

```
            TMRD(I)=TRD(I)
            FRNVP(I)=(TMRD(I)+460.0)*DZRD(I)
            FFRNVP(I)=DZRD(I)/(TMRD(I)+460.0)
            R(I)=RSTD(I)*(TMRD(I)+460.0)**2/(TR+460.0)**2
         ENDIF
         IF (IBTN(I).EQ.0) GO TO 30
         IF (NSAC(I,2).EQ.0) THEC
            F1=ABS(TJS(I)-TROCK(I))
            F2=ABS(TRD(I)-TROCK(I))
            FF=MAX(F1,F2)
            IF (FF.LT.0.1) THEN
               TMRD(I)=TJS(I)
               TMSQR=(TJS(I)+460.0)**2
            ELSE
               TTROCK=TROCK(I)
               IF (ABS(TJS(I)-TTROCK).LT.0.01) TTROCK=TJS(I)-5.0
               TT=(TRD(I)-TTROCK)/(TJS(I)-TTROCK)
               IF (TT.LT.0.01) THEN
                  TTROCK=MIN(TJS(I),TRD(I))-5.0
                  TT=(TRD(I)-TTROCK)/(TJS(I)-TTROCK)
               ENDIF
               IF (ABS(TT-1.0).LT.1.E-3) THEN
                  TMRD(I)=TJS(I)
                  TMSQR=(TJS(I)+460.0)**2
               ELSE
                  C=TJS(I)-TTROCK
                  B=ALOG(1.0/TT)/LA(I)
C
C     TMRD(I): MEAN AIR TEMP. IN AIRWAY NO(I).
C     TMSQR: MEAN-SQUARE TEMP. IN THE SAME AIRWAY.
C
                  TMRD(I)=C*(1.0-TT)/(B*LA(I))+TTROCK
                  TMSQR=(C**2)*(1.0-TT**2)/(2*B*LA(I))+2.0*(TTROCK+
     .                    460.)*C*(1.0-TT)/(B*LA(I))+(TTROCK+460.0)**2
               ENDIF
            ENDIF
         ELSE
C
            NZ=NSAC(I,2)
            IQ=IBTN(I)
            TMRD(I)=0.0
            TMSQR=0.0
            DO 10 J=1,NZ+1
               IF (J.EQ.1) THEN
                  D1=0.0
                  D2=LA(I)-DPPA(IQ,J,1)
                  C1=TAUXC(IQ)
                  C2=TRD(I)
               ELSE
                  D1=0.0
                  D2=DPPA(IQ,J-1,1)-DPPA(IQ,J,1)
                  C1=TAUXA(IQ,J-1)
                  C2=DPPA(IQ,J-1,2)
```

```
                ENDIF
                IF (D2.LT.1.E-5) D2=0.01
                F1=ABS(C1-TROCK(I))
                F2=ABS(C2-TROCK(I))
                FF=MAX(F1,F2)
                IF (FF.LT.0.1) THEN
                   TMRD(I)=TMRD(I)+C1*D2
                   TMSQR=TMSQR+D2*(C1+460.0)**2
                ELSE
                   TTROCK=TROCK(I)
                   IF (ABS(C1-TTROCK).LT.0.01) TTROCK=C1-5.0
                   TT=(C2-TTROCK)/(C1-TTROCK)
                   IF (TT.LT.0.01) THEN
                      TTROCK=MIN(C1,C2)-5.0
                      TT=(C2-TTROCK)/(C1-TTROCK)
                   ENDIF
                   IF (ABS(TT-1.0).LT.1.E-3) THEN
                      TMRD(I)=TMRD(I)+C1*D2
                      TMSQR=TMSQR+D2*(C1+460.0)**2
                   ELSE
                      C=C1-TTROCK
                      B=ALOG(1.0/TT)/D2
                      TMRD(I)=TMRD(I)+C*(1.0-TT)/B+TTROCK*D2
                      TMSQR=TMSQR+(C**2)*(1.0-TT**2)/(2*B)+2.0*(TTROCK
                            +460.0)*C*(1.0-TT)/B+D2*(TTROCK+460.0)**2
                   ENDIF
                ENDIF
10           CONTINUE
             TMRD(I)=TMRD(I)/LA(I)
             TMSQR=TMSQR/LA(I)
          ENDIF
C
C    FRNVP(I): PRODUCT OF MEAN TEMP. AND ELEVATION VARIATION.
C
          FRNVP(I)=(TMRD(I)+460.0)*DZRD(I)
          FFRNVP(I)=DZRD(I)/(TMRD(I)+460.0)
C
          IF (NWTYP(I).NE.1) R(I)=RSTD(I)*TMSQR/(TR+460.0)**2
30   CONTINUE
C
     RETURN
     END
C
C
     SUBROUTINE  DISP (MARKX,ISQC,IBB)
C
C    ------------------------------------------------------------------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       1) DISCARDING THE TENTATIVELY OBTAINED DATA IN THE PREVIOUS
C          ITERATION TO AVOID UNNECESSARY RECORDS.
C       2) RESTORATION OF INITIAL CONDITIONS EXCEPT FOR AIRFLOW RATES.
C
C    ------------------------------------------------------------------------------------------------------------------------
```

```
C
$INCLUDE:'CMMN.DAT'
C
C    ABANDON THE DATA FOR AIRWAYS WHICH GOT THEIR DETAILED DATA RECORDS
C    IN THE PREVIOUS ITERATION.
C
     DO 40 LRW=1,NB
        IF (IBTN(LRW).GT.0.AND.NSACB(LRW,1).EQ.0) THEN
           ISQCX=IBTN(LRW)
           DO 10 IV=1,NB
              IF (IBTN(IV).GE.ISQCX) THEN
                 IBTN(IV)=IBTN(IV)-1
              ENDIF
10         CONTINUE
           DO 20 IV=ISQCX,ISQC
              TAUXC(IV)=TAUXC(IV+1)
              DO 20 IU=1,10
                 TAUXA(IV,IU)=TAUXA(IV+1,IU)
                 DO 20 IW=1,5
                    DPPA(IV,IU,IW)=DPPA(IV+1,IU,IW)
20         CONTINUE
           IBTN(LRW)=0
           NSAC(LRW,1)=0
           NSAC(LRW,2)=0
           ISQC=ISQC-1
        ENDIF
40   CONTINUE
C
C    RESTORATION OF INITIAL CONDITIONS FOR THE PRESENT TIME INTERVAL
C    EXCEPT FOR AIRFLOW DISTRIBUTION.
C
     DO 60 I=1,NB
        TJS(I)=TTJS(I)
        TRD(I)=TTRD(I)
        RDPROP(I)=RDOP(I)
        RDCH4(I)=RCH4(I)
C
C    Q(I): AIRFLOW RATE IN AIRWAY NO(I) AT THE END OF THE PRESENT
C          INTERVAL.
C    QTP(I): INITIAL AIRFLOW RATE IN AIRWAY NO(I), CONSTANT WITHIN
C            EACH TIME INCREMENT.
C    QQ(I): TIME AVERAGED AIRFLOW RATE IN AIRWAY NO(I).
C
        QQA=0.55*Q(I)+0.45*QTP(I)
        QQ(I)=0.85*QQA+0.15*QQ(I)
        IF (CONCT.EQ.1) QQ(I)=QQQ(I)
        NSAC(I,1)=NSACB(I,1)
        NSAC(I,2)=NSACB(I,2)
        IQ=IBTN(I)
        IF (IQ.GT.0) THEN
           TAUXC(IQ)=TAUXD(IQ)
           DO 50 J=1,10
              TAUXA(IQ,J)=TAUXB(IQ,J)
              DO 50 JJ=1,5
```

```
                DPPA(IQ,J,JJ)=DPPB(IQ,J,JJ)
50          CONTINUE
          ENDIF
60    CONTINUE
      DO 65 I=1,NJ
          T(I)=TAS(I)
          PROP(I)=PROPS(I)
          PRCH4(I)=PRCH4S(I)
65    CONTINUE
      DO 70 I=1,IMX
          HTAD(I)=0.0
70    CONTINUE
      IBB=0
C
      CALL ARR (MARKX)
      CALL BASE
      CALL MSLIST
C
      RETURN
      END
C
C
      SUBROUTINE EVHT (MARKX,MARKP,MARKL,MARKR,QSUM,QS)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       1) DATA REARRANGEMENT FOR AIRWAYS IN WHICH AIRFLOW REVERSAL
C          HAPPENED IN THE PRESENT TIME INCREMENT.
C       2) DATA STORAGE ARRAY CONDENSATION FOR ARRAYS WHOSE CAPACITIES
C          ARE SATURATED.
C       3) UPDATING BACKUP ARRAYS TO SET UP THE INITIAL CONDITIONS FOR
C          THE NEXT TIME INCREMENT.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
      CP=0.24
      MARKP=1
      TACC=TACC+DELTAT
C
      CALL DTTR (MARKL,MARKP)
      IF (MARKL.NE.0) THEN
         CALL ARR (MARKX)
         CALL BASE
         CALL MSLIST
      ENDIF
C
C     CONDENSATION OF ARRAY DPPA AS ITS CAPACITY IS SATURATED FOR AN
C     AIRWAY. IT IS DONE BY ABANDONING SOME OF THE MOST CLOSELY LOCATED
C     DATA POINTS.
C
      DO 90 I=1,NB
```

```
      IQ=IBTN(I)
      IF (IQ.NE.0) THEN
         NS=NSAC(I,2)
         IF (NS.GE.9) THEN
            DO 84 JB=1,2
               JG=NSAC(I,2)
               SS1=1.E5
               JEE=1
               DO 70 JD=1,JG-1
                  SS2=DPPA(IQ,JD,1)-DPPA(IQ,JD+1,1)
                  IF (SS2.LT.SS1) THEN
                     SS1=SS2
                     JEE=JD
                  ENDIF
70             CONTINUE
               IF (JEE.EQ.1) THEN
                  JE=2
               ELSE
                  SS1=DPPA(IQ,JEE-1,1)-DPPA(IQ,JEE,1)
                  SS2=DPPA(IQ,JEE+1,1)-DPPA(IQ,JEE+2,1)
                  JE=JEE
                  IF (SS1.GE.SS2) JE=JEE+1
               ENDIF
               NSAC(I,2)=NSAC(I,2)-1
               NX=NSAC(I,2)
               TAUXA(IQ,JE-1)=TAUXA(IQ,JE)
               DPPA(IQ,JE-1,5)=DPPA(IQ,JE-1,5)+DPPA(IQ,JE,5)
               DO 82 JC=JE,NX+1
                  IF (JC.LE.NX) THEN
                     TAUXA(IQ,JC)=TAUXA(IQ,JC+1)
                     DO 80 JD=1,5
                        DPPA(IQ,JC,JD)=DPPA(IQ,JC+1,JD)
80                   CONTINUE
                  ELSE
                     TAUXA(IQ,JC)=0.0
                     DO 81 JD=1,5
                        DPPA(IQ,JC,JD)=0.0
81                   CONTINUE
                  ENDIF
82             CONTINUE
84          CONTINUE
         ENDIF
      ENDIF
90    CONTINUE
C
C     QSUM: SUM OF ABS. AIRFLOW VARIATION IN AIRWAYS, CRITERION FOR
C           OUTPUT DURING DYNAMIC SIMULATION.
C     DELTAQ(I): AIRFLOW VARIATION FOR AIRWAY NO(I) BETWEEN TWO
C                SUCCESSIVE OUTPUT.
C     MARKR=0: REGULAR OUTPUT WAS GIVEN IN THE LAST TIME INTERVAL.
C              DELTAQ(I) MUST BE INITIATED TO ZERO.
C     MARKR=1: THE OUTPUT FOR THE PRESENT INTERVAL HAS BEEN OMITTED.
C              DELTAQ(I) HAS ITS CUMULATIVE EFFECT.
C
```

```
      QS=QSUM
      QSUM=0.0
      DO 110 I=1,NB
         NSACB(I,1)=NSAC(I,1)
         NSACB(I,2)=NSAC(I,2)
         IF (MARKR.EQ.0) DELTAQ(I)=0.0
         DELTAQ(I)=DELTAQ(I)+Q(I)-QTP(I)
         QSUM=QSUM+ABS(DELTAQ(I))
         IF (IBTN(I).NE.0) THEN
            IQ=IBTN(I)
            TAUXD(IQ)=TAUXC(IQ)
            DO 100 J=1,10
               TAUXB(IQ,J)=TAUXA(IQ,J)
               DO 100 K=1,5
                  DPPB(IQ,J,K)=DPPA(IQ,J,K)
100         CONTINUE
         ENDIF
110   CONTINUE
      QS=ABS(QS-QSUM)
C
      RETURN
      END
C
C
      SUBROUTINE OUTPUT (IPT,MARKX,MARKY,ITT)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSE:
C        OUTPUT OF NORMAL CALCULATION RESULTS.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
C
      DIMENSION LST(NMX),TQR(NMX)
C
C
      IF (IPT.NE.1) THEN
         PP=0.0
         DO 10 L=1,NB
            IF (NWTYP(L).LE.0.OR.NWTYP(L).EQ.10) THEN
               P(L)=R(L)*Q(L)*ABS(Q(L))*1.E-10
            ELSE
               P(L)=R(L)
            ENDIF
            PP=PP+P(L)
10       CONTINUE
         PCRIT=PP/FLOAT(NB)
C
C
         IF (ITT.GE.ITN.AND.IPT.EQ.4) THEN
            DO 1520 I=1,NB
               TTMN=0.0
```

```
            DO 1500 J=1,10
                TTMN=TTMN+TMRCD(I,J)
1500        CONTINUE
            TTMN=TTMN/10.0
            TQR(I)=0.0
            DO 1510 J=1,10
                TQR(I)=TQR(I)+(TMRCD(I,J)-TTMN)**2
1510        CONTINUE
1520    CONTINUE
        IZ=0
        LL=0
1530    II=0
        DO 1540 I=1,NB
            IF (NO(I).GT.0) THEN
               IF (TQR(I).GT.TSR.OR.II.EQ.0) THEN
                  TSR=TQR(I)
                  II=I
               ENDIF
            ENDIF
1540    CONTINUE
        IF (II.EQ.0) GO TO 1590
        IF (LL.GT.0) THEN
           KK=0
           DO 1550 J=1,LL
               K=LST(J)
               IF (TQR(II).GT.TQR(K)) KK=KK+1
1550       CONTINUE
           IF (KK.GE.(LL-2).AND.TQR(II).GT.20.0) GO TO 1560
           GO TO 1590
        ENDIF
1560    LL=LL+1
        LST(LL)=II
        NO(II)=-NO(II)
        JSS=JS(II)
        JFF=JF(II)
1570    DO 1580 I=1,NB
            IF (NO(I).GT.0) THEN
               IF ((JS(I).EQ.JSS.OR.JF(I).EQ.JSS.OR.JS(I).EQ.JFF.OR.
                  JF(I).EQ.JFF).AND.(TQR(I).GT.20.0)) THEN
                  LL=LL+1
                  LST(LL)=I
                  NO(I)=-NO(I)
                  IF (JS(I).EQ.JSS) THEN
                     JSS=JF(I)
                  ELSE IF (JF(I).EQ.JSS) THEN
                     JSS=JS(I)
                  ELSE IF (JS(I).EQ.JFF) THEN
                     JFF=JF(I)
                  ELSE
                     JFF=JS(I)
                  ENDIF
                  GO TO 1570
               ENDIF
            ENDIF
```

```
130


1580        CONTINUE
            IZ=IZ+1
            IF (IZ.LE.3) GO TO 1530
1590        CRT1=0.0
            CRT2=0.0
            CRT3=0.0
            DO 1630 I=1,LL
               K=LST(I)
               NO(K)=IABS(NO(K))
               CRT=0.0
               DO 1600 J=10,3,-1
                  CRT=CRT+(TMRCD(K,J)-TMRCD(K,J-2))**2
1600           CONTINUE
               CRT1=CRT1+CRT/8.0
               CRT=0.0
               DO 1610 J=10,4,-1
                  CRT=CRT+(TMRCD(K,J)-TMRCD(K,J-3))**2
1610           CONTINUE
               CRT2=CRT2+CRT/7.0
               CRT=0.0
               DO 1620 J=10,5,-1
                  CRT=CRT+(TMRCD(K,J)-TMRCD(K,J)-TMRCD(K,J-4))**2
1620           CONTINUE
               CRT3=CRT3+CRT/6.0
1630        CONTINUE
            CRT1=CRT1/LL
            CRT2=CRT2/LL
            CRT3=CRT3/LL
            LSG=LL
            IF (MIN(CRT1,CRT2,CRT3)*10.0.LE.MAX(CRT1,CRT2,CRT3)) THEN
               ICYCLE=2
               IF (CRT2.LT.0.95*CRT1) ICYCLE=3
               IF (CRT3.LT.0.9*CRT1.AND.CRT3.LT.CRT2) ICYCLE=4
            ELSE
               ICYCLE=0
            ENDIF
         ENDIF
      ENDIF
      IF (IPT.EQ.4) THEN
C
C
         IF (IAC.GT.0) THEN
            WRITE (8,860)
            DO 76 IR=1,IAC
               II=IOMIT(I)
               WRITE (8,870) NO(II),JS(II),ROMIT(IR,1),ROMIT(IR,2),
                             ROMIT(IR,3)
76          CONTINUE
            IAC=0
         ENDIF
         ITL=ITT
         JTL=ITN
         WRITE (8,320) TACC
         IF (ITL.GE.JTL) THEN
```

```
      WRITE (8,690) SUMFNV/MNO
      KTL=0
      DO 78 I=1,NB
         NREVA=NREV(I)
         NREVA=NNREV(I)
         IF (NREVA.GE.ITN/3) THEN
            IF (KTL.EQ.0) THEN
               WRITE (8,700)
               KTL=1
            ENDIF
            IF (NWTYP(I).EQ.1) THEN
               PP=R(I)
            ELSE
               PP=R(I)*Q(I)*ABS(Q(I))*1.E-10
            ENDIF
            WRITE (8,710) NO(I),Q(I),PP,NREVA,ITL
         ENDIF
78    CONTINUE
      IZ=ITN-9
      IF (ICYCLE.EQ.0) THEN
         WRITE (8,890) (IJ,IJ=IZ,IZ+9)
      ELSE
         WRITE (8,900) ICYCLE,(IJ,IJ=IZ,IZ+9)
      ENDIF
      WRITE (8,905) (HRM(IJ),IJ=1,10)
      DO 80 I=1,LSG
         K=LST(I)
         IF (JSB(K).EQ.JS(K)) THEN
            JSS=JS(K)
            JFF=JF(K)
         ELSE
            JSS=JF(K)
            JFF=JS(K)
         ENDIF
         WRITE (8,910) NO(K),JSS,JFF,(TMRCD(K,J),J=1,10)
         WRITE (8,920) (QRCD(K,J)/1000.0,J=1,10)
80    CONTINUE
   ELSE
      IF (SUMFNV/MNO.LE.2.E-4) WRITE (8,770) SUMFNV/MNO,ITL
      IF (SUMFNV/MNO.GT.2.E-4) WRITE (8,780) SUMFNV/MNO,ITL
   ENDIF
   LINE=NCOMT1+1
   WRITE (8,750)
   DO 86 I=1,NB
85       IJ=0
         IF (LINE.LE.50) IJ=ITITLE(LINE)
         IF (I.EQ.IJ) THEN
            IF (LINE.EQ.1) I1=0
            IF (LINE.NE.1) I1=ITITLE(LINE-1)
            IF (LINE.EQ.50) I2=0
            IF (LINE.NE.50) I2=ITITLE(LINE+1)
            IF (I2.NE.ITITLE(LINE).AND.I1.NE.ITITLE(LINE)) THEN
               WRITE (8,930) (TITLE(LINE,J),J=1,18)
            ELSE IF (I2.EQ.ITITLE(LINE).AND.I1.EQ.ITITLE(LINE)) THEN
```

```
            WRITE (8,960) (TITLE(LINE,J),J=1,18)
          ELSE IF (I1.EQ.ITITLE(LINE)) THEN
            WRITE (8,940) (TITLE(LINE,J),J=1,18)
          ELSE
            WRITE (8,950) (TITLE(LINE,J),J=1,18)
          ENDIF
          LINE=LINE+1
          GO TO 85
        ENDIF
        IF (CONCT.EQ.0) THEN
        WRITE (8,760) NO(I),IABS(JS(I)),IABS(JF(I)),DELTAQ(I),Q(I),
     .                TMRD(I),TRD(I),RDPROP(I),RDCH4(I),P(I)
        ELSE
        WRITE (8,765) NO(I),IABS(JS(I)),IABS(JF(I)),DELTAQ(I),Q(I),
     .                                    RDPROP(I),RDCH4(I),P(I)
        ENDIF
86      CONTINUE
        LI=0
        DO 87 I=1,NB
          IF (NSAC(I,1).EQ.1) THEN
            LK=IBTN(I)
            LM=0
            IF (DPPA(LK,2,3).GE.1.E-4) LM=2
            IF (DPPA(LK,1,3).GE.1.E-4) LM=1
            IF (LM.GT.0) THEN
              IF (LI.NE.1) THEN
                LI=1
                WRITE (8,350)
              ENDIF
              IF (CONCT.EQ.0) THEN
                WRITE (8,360) NO(I),DPPA(LK,LM,1),JS(I),
     .                DPPA(LK,LM,2),DPPA(LK,LM,3),DPPA(LK,LM,4)
              ELSE
                WRITE (8,365) NO(I),DPPA(LK,LM,1),JS(I),
     .                              DPPA(LK,LM,3),DPPA(LK,LM,4)
              ENDIF
            ENDIF
          ENDIF
87      CONTINUE
        IF (INFLOW.GT.0) THEN
          WRITE (8,820)
          DO 89 L=1,INFLOW
            DO 88 J=1,NB
              IF (NCENT(L).EQ.NO(J)) THEN
                WRITE (8,830) NO(J),Q(J),TJS(J),RDPROP(J),
     .                  O2BEH(L),HTAD(L)
                IF (TJS(J).GT.3200.0) WRITE (8,995) NO(J)
                GO TO 89
              ENDIF
88          CONTINUE
89        CONTINUE
        ENDIF
        IF (IOUT.NE.0) THEN
          WRITE (8,650)
```

```
          IF (CONCT.EQ.0) THEN
            WRITE (8,660) (JNO(I),T(I),PROP(I),PRCH4(I),I=1,NJ)
          ELSE
            WRITE (8,665) (JNO(I),     PROP(I),PRCH4(I),I=1,NJ)
          ENDIF
        ENDIF
        KTL=0
        DO 90 I=1,NB
          NNREVA=NNREV(I)
          IF ((NNREVA/2)*2.NE.NNREVA) THEN
            IF (KTL.EQ.0) THEN
              WRITE (8,470)
              KTL=1
            ENDIF
            WRITE (8,480) NO(I),IABS(JS(I)),IABS(JF(I))
          ENDIF
90      CONTINUE
      ELSE IF (IPT.EQ.5) THEN
        WRNSUM=WRNPR+WRNGS+WRNSM+WRNHT
        IF (WRNSUM.LE.0.) THEN
          WRITE (8,410)
        ELSE
          J=0
          WRNP=WRNPR
          IF (PCRIT*0.05.LE.0.5*WRNPR) WRNP=PCRIT*0.05
          DO 95 I=1,NB
            K=0
            L=0
            M=0
            N=0
            IF ((100.*RDCH4(I)).GE.WRNGS) K=K+1
            IF ((100.*RDPROP(I)).GE.WRNSM) L=L+1
            IF (TRD(I).GE.WRNHT) M=M+1
            IF (P(I).LT.WRNP) N=N+1
            IF ((K+L+M+N).GT.0) THEN
              IF (J.EQ.0) THEN
                WRITE (8,420) WRNGS,WRNSM,WRNHT,WRNP
                J=J+1
              ENDIF
              WRITE (8,430) NO(I),IABS(JS(I)),IABS(JF(I)),RDCH4(I),
                            RDPROP(I),TRD(I),P(I)
            ENDIF
95        CONTINUE
          IF (J.LE.0) WRITE (8,440)
          J=0
          KM=0
          DO 100 I=1,NJ
            K=0
            L=0
            M=0
            IF ((100.*PRCH4(I)).GE.WRNGS) K=K+1
            IF ((100.*PROP(I)).GE.WRNSM) L=L+1
            IF (T(I).GE.WRNHT) M=M+1
            IF ((K+L+M).GT.0) THEN
```

```
                IF (J.EQ.0) THEN
                   WRITE (8,450) WRNGS,WRNSM,WRNHT,WRNGS,WRNSM,WRNHT
                ENDIF
                J=J+1
                IF (2*(J/2).NE.J) THEN
                   KM=I
                ELSE
                   WRITE (8,460) JNO(KM),PRCH4(KM),PROP(KM),T(KM),
     .                           JNO(I),PRCH4(I),PROP(I),T(I)
                   KM=0
                ENDIF
             ENDIF
100       CONTINUE
          IF (KM.GT.0) WRITE (8,460) JNO(KM),PRCH4(KM),PROP(KM),T(KM)
          IF (J.LE.0) WRITE (8,445)
       ENDIF
    ELSE IF (IPT.EQ.6) THEN
       IF (CONCT.EQ.1) GO TO 107
       NRW=0
       DO 105 L=1,NB
          LL=IBTN(L)
          IF (LL.GT.0) THEN
             IF (NRW.EQ.0) THEN
                WRITE (8,570)
                NRW=1
             ENDIF
             LL1=NSAC(L,2)
             IF (LL1.GT.0) THEN
                WRITE (8,580) NO(L),(DPPA(LL,JQ,1),JS(L),DPPA(LL,JQ,2)
     .                        ,TAUXA(LL,JQ),(DPPA(LL,JQ,JP),JP=3,5),
     .                        JQ=LL1,1,-1)
             ELSE
                WRITE (8,640) NO(L),TJS(L),TRD(L),RDPROP(L)
             ENDIF
          ENDIF
105    CONTINUE
107    CONTINUE
    ENDIF
C
    RETURN
C
110 FORMAT (///,22X,'NETWORK CALCULATION RESULTS ',/)
120 FORMAT (1X,'AIRWAY',2X,'FROM',4X,'TO',3X,'TYPE',8X,'RESISTANCE',
     .          7X,'AIRFLOW',6X,'HEADLOSS',/)
130 FORMAT (I5,I7,I6,I7,8X,E9.4,6X,F10.0,3X,F10.3)
140 FORMAT (///,1X,'NETWORK CONTAINS NO FANS')
150 FORMAT (///,T20,'FANS'//' AIRWAY  FROM    TO     AIRFLOW    FAN',
     .          ' PRESSURE',/)
160 FORMAT (I5,I7,I7,F13.0,F13.3)
170 FORMAT (/////,' THESE CHARACTERISTICS WERE STORED FOR FANS',5I5,/
     .          T45,5I5)
175 FORMAT (//,5(' Q*1000   PF '),/)
180 FORMAT (5(F8.1,F6.2)/,5(F8.1,F6.2),/)
190 FORMAT (///,1X,'THE STATED NUMBER OF AIRWAYS WAS',I9,/,1X,'THE ',
```

```
     .          'STATED NUMBER OF JUNCTIONS WAS',I7)
 200 FORMAT (///,1X,'FAN CHARACTERISTICS ARE EXCEEDED FOR FAN NO.',I5)
 240 FORMAT (///,1X,'A TEMPERATURE OF',F6.1,' WAS ASSIGNED TO ',
     .          'JUNCTION NO.',I7)
 250 FORMAT (/////,T18,'THE FOLLOWING CONTAMINATION WAS ASSUMED',//,
     .          12X,'FIXED INFLUX',T27,'O2 CONCENTRATION',T44,
     .          'PRODUCTION PER FT3 O2',T67,'REF. Q',/,' AIRWAY  FLUX  %',
     .          'CONC',3X,'HEAT',3X,'BEHIND FIRE',5X,'FUMES',5X,'HEAT',/)
 260 FORMAT (I5,F8.1,F6.2,1X,E9.3,3X,F6.2,4X,F8.2,F10.1,I12)
 270 FORMAT (///,1X,' NO CONTAMINATION WAS SPECIFIED')
 320 FORMAT (///,20X,'TIME AT ',F6.0,' SEC. AFTER EVENT',//)
 350 FORMAT (///,T21,'DATA FOR THE FUME FRONT IN AIRWAYS',
     .          ///,1X,'AIRWAY',T11,'POSITION',T21,'FROM',
     .          T37,'TEMPERATURE',T51,'FUME',T61,'METHANE',/)
 360 FORMAT (1X,I5,T11,0PF7.2,T19,I5,T38,0PF8.2,T49,2PF6.2,T61,2PF6.2)
 365 FORMAT (1X,I5,T11,0PF7.2,T19,I5,          T49,2PF6.2,T61,2PF6.2)
 410 FORMAT (///,1X,'NO THRESHOLD FOR CRITICAL STATES ',
     .          'WERE SPECIFIED.')
 420 FORMAT (///,9X,'IN THE FOLLOWING AIRWAYS EXIST CRITICAL ',
     .          'CONDITIONS',//,1X,'AIRWAY',T10,'FROM',T16,'TO',T22,'CH4 %'
     .          ,T33,'FUMES %',T45,'TEMPERATURE',T61,'HEADLOSS',/,
     .          T22,'>',F5.3,T33,'>',F6.3,T46,'>',F5.0,' F',T59,
     .          '<',F6.3,' IN.WG.',/)
 430 FORMAT (I5,I7,I5,T21,2PF6.2,T32,2PF8.3,T45,0PF8.1,T62,0PF6.3)
 435 FORMAT (I5,I7,I5,T21,2PF6.2,T32,2PF8.3,          T62,0PF6.3)
 440 FORMAT (///,1X,'NO CRITICAL CONDITIONS AT AIRWAY ENDS ',
     .          'WERE DETECTED')
 445 FORMAT (///,1X,'NO CRITICAL CONDITIONS IN JUNCTIONS WERE ',
     .          'DETECTED.')
 450 FORMAT (///,10X,'IN THE FOLLOWING JUNCTIONS EXIST CRITICAL ',
     .          'CONDITIONS',//,' JUNCTION',T11,'CH4  %',T19,'FUMES %',T28
     .          ,'TEMP. F',T40,'JUNCTION',T49,'CH4  %',T57,'FUMES %',T66,
     .          'TEMP. F',/,T11,'>',F5.2,T19,'>',F6.3,T29,'>',F5.0,T49,
     .          '>',F5.2,T57,'>',F6.3,T67,'>',F5.0,/)
 460 FORMAT (I6,T11,2PF5.2,T17,2PF8.2,T27,0PF7.1,T39,I6,T49,2PF5.2,
     .          T55,2PF8.2,T65,0PF7.1)
 465 FORMAT (I6,T11,2PF5.2,T17,2PF8.2,          T39,I6,T49,2PF5.2,
     .          T55,2PF8.2          )
 470 FORMAT (///,1X,'REVERSAL OF AIRFLOW HAS OCCURRED IN THE ',
     .          'FOLLOWING PLACES',/)
 480 FORMAT (1X,'AIRWAY',I6,' IS NOW CARRYING AIR FROM',I6,' TO',I6)
 490 FORMAT (/////,3X,'NO NETWORK CALCULATION HAS BEEN PERFORMED, ',
     .          'THESE ARE THE INPUT DATA',//,' AIRWAY FROM',T15,'TO',
     .          T22,'AIRFLOW',T30,'TYPE',T37,'LENGTH',T47,'AREA',
     .          T57,'R',T64,'K',T68,'PERI.',/)
 500 FORMAT (I5,I6,T13,I4,T19,F10.0,T30,I3,T36,F7.1,T44,F7.1,T53,E8.3,
     .          T62,I4,F7.1)
 570 FORMAT (///,23X,'DATA RECORD IN AIRWAYS',//,1X,'AIRWAY',T12,
     .          'DIST.',T19,'FROM',T26,'FRONT T',T37,'REAR T',
     .          T47,'FUME',T57,'CH4',T63,'AIR M (LB)',/)
 580 FORMAT (1X,I5,11(T8,0PF9.2,T17,I5,T24,0PF9.2,T34,0PF9.2,T44,
     .          2PF7.2,T53,2PF7.2,T61,0PF9.2,/))
 640 FORMAT (1X,I5,T9,'NO C.V. T(BGG)=',0PF7.1,', T(END)=',0PF7.1,',
     .          FUME=',2PF5.2,'%.',/)
```

```
650  FORMAT (///,19X,'PARAMETERS OF AIR IN JUNCTIONS',//,1X,'JUNCTION'
     .       ,T13,'TEMP.',T20,'FUMES',T27,'METHANE',T42,'JUNCTION',T53,
     .       'TEMP.',T60,'FUMES',T67,'METHANE',/)
660  FORMAT (I6,T9,0PF9.2,T18,2PF7.2,T26,2PF7.2,T41,I6,T49,0PF9.2,T58,
     .       2PF7.2,T66,2PF7.2)
665  FORMAT (I6,        T18,2PF7.2,T26,2PF7.2,T41,I6,        T58,
     .       2PF7.2,T66,2PF7.2)
690  FORMAT (///,1X,'THE CALCULATION WAS NOT COMPLETED SINCE THE ',
     . .     'NUMBER OF EXCHANGES BETWEEN',/,' NETWORK AND TEMPERATURE'
     .       ,' PARTS OF THE PROGRAM BECAME EXCESSIVE.',/,' THRESHOLD',
     .       ' IN NVP IS 0.0002 IN.W.G. PER MESH. THE NVP CORRECTION ',
     .       'PER'/,' MESH NOW EQUALS ',F7.5,' IN.W.G.')
700  FORMAT (///,1X,'THE FOLLOWING UNSTABLE AIRWAYS MAY HAVE',1X,
     .       'CAUSED THE INCOMPLETION',/,' OF NETWORK BALANCING:',/)
710  FORMAT (1X,'AIRWAY',I5,' OF AIRFLOW RATE',F7.0,' (CFM) AND ',
     .       'HEADLOSS',F7.3,' IN.W.G.',/,' HAD ITS AIRFLOW DIRECTION'
     .       ,' REVERSED',I3,' TIMES IN',I3,' ITERATIONS')
750  FORMAT (///,T8,'TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN'
     .       ,' AIRWAYS',//,' AIRWAY',T9,'FROM',T15,'TO',T19,'DELTA Q',
     .       T29,'AIRFLOW',T38,'AVE. T',T45,'T AT END',T54,'FUMES',T61,
     .       'CH4',T65,'HEADLOSS',/)
760  FORMAT (I5,I6,I5,T17,0PF8.0,T27,0PF9.0,T37,0PF7.2,T45,0PF7.2,T52,
     .       2PF6.2,T58,2PF6.2,T66,0PF6.3)
765  FORMAT (I5,I6,I5,T17,0PF8.0,T27,0PF9.0,                T52,
     .       2PF6.2,T58,2PF6.2,T66,0PF6.3)
770  FORMAT (/////,' THRESHOLD IN ACCURACY (SUM OF NVP CORRECTIONS ',
     .       'PER MESH < 2.E-4 IN.W.G.)',/,' SATISFIED. CURRENT ',
     .       'SUMFNV PER MESH',F9.6,' IN.W.G.,  ITERATIONS ',I5)
780  FORMAT (/////,1X,'THRESHOLD IN ACCURACY (SUM OF NVP',1X,
     .       'CORRECTIONS PER MESH<2.E-4 IN. W.G.)',/,' NOT ',
     .       'SATISFIED. CURRENT SUMFNV:',F9.6,' IN.W.G., ITER.',I5)
800  FORMAT (/////,12X,'DATA FOR CONCENTRATION AND TEMPERATURE',
     .       ' CALCULATION',//,1X,'AIRWAY  FROM',T16,'TO',T21,
     .       'ELEV. DIFF.',T34,'ROCK T',T43,'CH4 PROD.',T54,'COND.',
     .       T61,'DIFFUSIVITY',/)
810  FORMAT (I5,I7,I5,T21,F9.2,T33,F7.2,T43,F7.1,T52,F7.2,T61,F8.3)
820  FORMAT (///,23X,'DATA FOR FIRE SOURCES',//,1X,'AIRWAY',T11,
     .       'AIRFLOW',T21,'TEMPERATURE',T37,'FUME',T51,'O2 LEFT',T61,
     .       'HEAT INPUT',/)
830  FORMAT (I5,T8,0PF10.0,T22,0PF8.2,T35,2PF6.2,T50,0PF7.2,T62,E8.3)
840  FORMAT (/,1X,'AIRFLOW REVERSED IN FAN BRANCH NO. ',I5)
860  FORMAT (//,1X,'THE FOLLOWING DATA RECORDS WERE OMITTED DUE TO ',
     .       'ARRAY CAPACITIES.',//,T1,'AIRWAY',T15,'PARAMETERS AT ',
     .       'STARTING JUNCTIONS IN THE CURRENT TIME INTERVAL',//,T22,
     .       'JS',T31,'TEMPERATURE',T46,'FUME %',T61,'CH4 %',/)
870  FORMAT (1X,I5,T19,I5,T31,F8.1,T44,F7.3,T59,F7.3)
890  FORMAT (///,1X,'THE FAILURE IN CONVERGING WAS NOT CAUSED BY A',
     .       ' CYCLIC PATTERN. DATA OF',/,' SIGNIFICANT CONDITION ',
     .       'CHANGE DURING LAST 10 ITERATIONS FOLLOW. FURTHER',/,
     .       ' ANALYSES ARE NEEDED. ("ITR" STANDS FOR ITERATION)',//,
     .       ' UNITS: Q: AIRFLOW RATE IN 1000FT3/MIN;',17X,'T: TEMP. ',
     .       'IN F;',/,8X,'H: NVP CORRECTION PER MESH IN IN.W.G.',//,
     .       ' ITR',T5,I5,9I7,/)
900  FORMAT (///,1X,'THE FAILURE IN CONVERGING WAS CAUSED BY A CYCLIC',
```

```
     .            ' PATTERN IN EVERY',I3,/,' ITERATIONS AS SHOWN BELOW:',
     .            ' ("ITR" STANDS FOR ITERATION)',//,' UNITS: Q: AIRFLOW ',
     .            'RATE IN 1000FT3/MIN;',17X,'T: TEMP. IN F;',/,8X,'H: NVP ',
     .            'CORRECTION PER MESH IN IN.W.G.',//,' ITR',T5,I5,9I7,/)
905  FORMAT (1X,'H',T3,10(1X,F6.4),/)
910  FORMAT (1X,'AIRWAY: ',I4,10X,'JS:',I4,6X,'JF:',I4,/,1X,
     .            'T',T3,10F7.1)
920  FORMAT (1X,'Q',T3,10F7.1,/)
930  FORMAT (/,18A4,/)
940  FORMAT (18A4,/)
950  FORMAT (/,18A4)
960  FORMAT (18A4)
970  FORMAT (/////,20X,'TIME AT ',F5.1,' HOUR(S) AFTER EVENT',//)
980  FORMAT (///,T2,'JUNCTION',T12,'TEMP.',T20,'ELEVATION',T32,'CH4',
     .            T40,'JUNCTION',T50,'TEMP.',T58,'ELEVATION',T70,'CH4',/)
990  FORMAT (T2,I5,T10,F7.1,T19,F8.1,T30,F5.2,T40,I5,T48,F7.1,T57,F8.1
     .            ,T68,F5.2)
995  FORMAT (/,'* WARNING *  AIR TEMP. IN FIRE',I5,' IS EXCEPTIONALLY',
     .            ' HIGH FOR MINE FIRE.',/,' THE POSSIBLE CAUSES ARE:',/,
     .            ' 1. "QCENT" WAS NOT SPECIFIED;',/,' 2. "QCENT" DOES NOT',
     .            ' MATCH "HEAT";',/,' 3. "HTPO2" WAS TOO LARGE.')
C
      END
C
C
      SUBROUTINE READIN (DAL,MAX,ISTOP,ICT)
C
C     ------------------------------------------------------------------------------------------------------------------------
C
C        SUBROUTINE PURPOSE:
C        DATA INPUT.
C
C     ------------------------------------------------------------------------------------------------------------------------
C
$INCLUDE:'CMMN.DAT'
      CHARACTER*1 CHK(15),ER(80)
      DIMENSION DAL(15)
      DATA CHK/'0','1','2','3','4','5','6','7','8','9','.',',','+',
     .            '-',' '/
C
C
      KK=0
      KK1=0
      KK2=0
      NEG=0
      ISTOP=0
      DO 5 I=1,15
          DAL(I)=0.0
5     CONTINUE
      DO 7 I=1,80
          ER(I)=' '
7     CONTINUE
      READ (9,110) (ROW(I),I=1,80)
      LLINE=LLINE+1
```

```
      IDEC=0
      MLEV=1
      JJ=0
      I=15
      LEV=0
10    LEV=LEV+1
      II=I
      IF (LEV.GT.80) GO TO 50
      DO 20 I=1,15
         IF (ROW(LEV).EQ.CHK(I)) THEN
            IF (I.NE.15) KK=1
            GO TO 30
         ENDIF
20    CONTINUE
      KK2=KK2+1
      IF ((KK.EQ.0.AND.ROW(LEV).NE.'$').OR.KK2.GE.4) THEN
         ISTOP=0
         NCOMTS=NCOMTS+1
         IF (NCOMTS.LE.50) THEN
            IF (ICT.EQ.1) THEN
               KK3=0
               KK4=0
               DO 21 I=1,80
                  IF (ROW(I).NE.CHK(15)) GO TO 22
                  KK3=KK3+1
21             CONTINUE
22             DO 23 I=80,1,-1
                  IF (ROW(I).NE.CHK(15)) GO TO 24
                  KK4=KK4+1
23             CONTINUE
24             LR=(KK3+KK4-8)/2
               IF (LR.LT.0) LR=0
               KK5=80-KK4
               IF (LR.EQ.0) KK5=KK3+72
               DO 25 I=KK3+1,KK5
                  ER(I-KK3+LR)=ROW(I)
25             CONTINUE
               DO 26 I=1,80
                  IF (I.LE.LR) THEN
                     ROW(I)=' '
                  ELSE
                     ROW(I)=ER(I)
                  ENDIF
26             CONTINUE
            ENDIF
            DO 27 I=1,20
               J=(I-1)*4+1
               TITLE(NCOMTS,I)=ROW(J)//ROW(J+1)//ROW(J+2)//ROW(J+3)
27          CONTINUE
         ENDIF
         RETURN
      ELSE IF (KK.EQ.0.AND.ROW(LEV).EQ.'$') THEN
         KK1=1
         GO TO 50
```

```
      ELSE IF (KK1.NE.1) THEN
        ER(LEV)=ROW(LEV)
        ISTOP=1
        I=1
      ENDIF
30    IF (I.GT.10) GO TO 40
      IF (JJ.EQ.1) MLEV=MLEV+1
      JJ=0
      IF (IDEC.EQ.0) THEN
        DAL(MLEV)=(DAL(MLEV)*10.0)+((I-1)*1.0)
      ELSE IF (IDEC.GT.0) THEN
        DAL(MLEV)=DAL(MLEV)+((I-1)*1.0)/(10.0**IDEC)
      ENDIF
      IF (IDEC.NE.0) IDEC=IDEC+1
      GO TO 10
40    IF (I.EQ.11) IDEC=1
      IF (I.EQ.14) NEG=1
      IF (I.EQ.12.OR.I.EQ.15) THEN
        IF (NEG.EQ.1) DAL(MLEV)=0.0-DAL(MLEV)
        NEG=0
        IDEC=0
      ENDIF
      IF (JJ.EQ.1.AND.(I.EQ.11.OR.I.EQ.13.OR.I.EQ.14)) THEN
        MLEV=MLEV+1
        JJ=0
      ENDIF
      IF (I.EQ.12) THEN
        MLEV=MLEV+1
        JJ=0
      ENDIF
      IF (I.EQ.15.AND.II.EQ.12) JJ=5
      IF (I.EQ.15.AND.II.NE.15.AND.II.NE.12.AND.JJ.EQ.0) JJ=1
      IF (I.EQ.12) DAL(MLEV)=0.0
      IF (MLEV.LE.MAX) GO TO 10
50    IF (ISTOP.GT.0) WRITE (8,100) (ROW(I),I=1,80),(ER(I),I=1,80)
      IF (KK.EQ.0.OR.KK1.EQ.1) DAL(15)=-1.E25
      RETURN
C
100   FORMAT (/,' * ERROR *  ILLEGAL CHARACTER DETECTED IN ',
     .        'THE LINE WHERE NUMERICAL INPUT',/,' WAS EXPECTED. ',
     .        'TWO LINES FOLLOW: 1 ORIGINAL; 2 ILLEGAL CHARACTERS.',
     .        //,80A1,/,80A1,/)
110   FORMAT (80A1)
C
      END


INCLUDE FILE

      PARAMETER
     .   ( NMX=500, NMY=350, NMZ=50, IMX=10, IMY=10, IMZ=10, LMX=10000 )
      REAL LA

      COMMON /CONTRL/ NB,NJ,NFNUM,NVPN,NETW,NTEMP,MADJ,ITN,DR,TR,TINC,
```

```
.               SPAN,IOUT,TOUT,CONCT
      COMMON /CTRL/     NAV,MAXJ,INFLOW,CRITSM,CRITGS,CRITHT
      COMMON /NTWK/     NO(NMX),JS(NMX),JF(NMX),NWTYP(NMX),R(NMX),Q(NMX),
     .          P(NMX),KF(NMX),LA(NMX),A(NMX),O(NMX),RSTD(NMX),
     .          DZRD(NMX),FRNVP(NMX),NREV(NMX),RDCH4(NMX),
     .          RDPROP(NMX),TRD(NMX),TJS(NMX),RDOP(NMX),RCH4(NMX),
     .          FFRNVP(NMX),RA(NMX),NNREV(NMX)
      COMMON /FAN/      NOF(IMX),NFREG(IMX),RGRAD(IMX),NFCW(IMX),
     .     MPTS(IMX),QF(IMX,IMY),PF(IMX,IMY),NSKP(IMX),NEGQ(IMX),JSB(NMX)
      COMMON /FC/       FC(IMY,IMY),DK(IMY),FK(IMY),FKQ(IMX,IMY,4)
      COMMON /MESH/     MNO,MEND(NMY),MSL(LMX),FNVP(NMY),RQ(NMX),INU(NMX),
     .          KNO(NMX),KJS(NMX),KJF(NMX)
      COMMON /JUNCT/    JNO(NMY),T(NMY),Z(NMY),CH4C(NMY),JNOL(NMY),
     .          PROP(NMY),PRCH4(NMY),JLR(NMY)

      COMMON /TEMP/     TROCK(NMX),TMRD(NMX),HA(NMX),HK(NMX)
      COMMON /FUME/     NCENT(NMX),CH4V(NMX),CH4PA(NMX),CONT(IMX),
     .          CONC(IMX),HEAT(IMX),O2MIN(IMX),SMPO2(IMX),HTPO2(IMX),
     .          TFSI(IMX),O2BEH(IMX),TPR(IMX),HTAD(IMX),QCENT(IMX)
      COMMON /SCHEME/ NGIN(NMX),MMIN(NMX),NGOUT(NMX),LOUT(NMX)
      COMMON /EST/      MEMREC(NMX),NOREC(NMX),ESTPR(NMX),ESTCH4(NMX),
     .          ESTTR(NMX)
      COMMON /FACTOR/ XNEW(NMX),DCOAGE(NMX)
      COMMON /TTJ/      TIME,TSTART,JSTART,TLEFT(NMX)
      COMMON /SUM/      SUMQ(NMY),SUMC(NMY),SUMM(NMY),SMHEAT(NMY)
      CHARACTER*4 TITLE
      CHARACTER*1 ROW

      COMMON /AUX/      PROPS(NMX),PRCH4S(NMX),QQ(NMX),QTP(NMX),TTJS(NMX),
     .          TTRD(NMX),TAS(NMX),BI(NMX)
      COMMON /WRN/      WRNPR,WRNGS,WRNSM,WRNHT,IOMIT(IMX),ROMIT(IMX,3),IAC
      COMMON /TRANS/    DELTAT,TACC,SUMFNV
      COMMON /RCD/      IBTN(NMX),DPPA(NMZ,10,5),NSAC(NMX,2),TMRDA(NMX),
     .          DELTAQ(NMX),DPPB(NMZ,10,5),NSACB(NMX,2),GBTN(10,5),
     .          JCH(NMY),TAUXA(NMZ,10),TAUXB(NMZ,10),TAUXC(NMZ),
     .          TAUXD(NMZ),FMASS(NMZ)
      COMMON /CCJ/      JDP,JDPP(IMX)
      COMMON /TREND/    QRCD(NMX,10),TMRCD(NMX,10),HRM(IMX)
      COMMON /CHRC/     ITITLE(50),NCOMT1,NCOMTS,NCOMT2,LLINE
      COMMON /ERMSG/    ROW(80),TITLE(50,20)
      COMMON /ATJUNT/   JAN,JAJ,JA1(10),JATP(10),JSTAR(10),TSTAR(10),
     .                  MSTAR(10),ISTAR(10)
      COMMON /CONCN/    TTT(NMY),TJSS(NMX),TRDD(NMX),QQQ(NMX)
```

APPENDIX B.–EXAMPLE OF MFIRE DATA FILE AND OUTPUT

Data file

```
$-----------------------------------------------------------------------------------------------------------------$
$                                                                                                                 $
$          *** INPUT FORMAT INSTRUCTION FILE FOR PROGRAM MFIRE  ***                                               $
$                                                                                                                 $
$   GENERAL RULES FOR DATA INPUT:                                                                                 $
$                                                                                                                 $
$  * THERE IS NO NEED TO DISTINGUISH INTERGER AND REAL NUMBERS.                                                   $
$  SYMBOLS FOR NUMERICAL INPUT INCLUDE THE FOLLOWING:                                                             $
$                                1 2 3 4 5 6 7 8 9 0 . , - + BLANK                                                $
$  * FREE FORMAT IS ALLOWED. COMMA AND BLANK SPACE CAN BOTH BE                                                    $
$  DELIMITERS. SEQUENTIALLY DISTRIBUTED BLANK SPACES ACT AS ONE                                                   $
$  DELIMITER. BLANKS ADJACENT TO A COMMA HAVE NO DELIMITING EFFECT.                                               $
$  * A ZERO INPUT CAN BE OMITTED BUT NOT THE DELIMITERS EXCEPT WHEN                                               $
$  THE ZERO INPUTS ARE THE LAST ONES ON A LINE.                                                                   $
$  * A LINE WHOSE FIRST NON-BLANK SYMBOL IS "$" IS AN INSTRUCTION                                                 $
$  LINE WHICH WILL BE IGNORED IN A PROGRAM RUN.                                                                   $
$  * A LINE WHOSE FIRST NON-BLANK SYMBOL IS NEITHER ONE USED IN THE                                               $
$  ABOVE LIST FOR NUMERICAL INPUT NOR "$", OR WHICH CONTAINS MORE                                                 $
$  THAN 3 SYMBOLS OUTSIDE OF THE LIST, IS A COMMENT LINE WHICH WILL                                               $
$  BE SHOWN IN THE OUTPUT FILE.                                                                                   $
$-----------------------------------------------------------------------------------------------------------------$
$-----------------------------------------------------------------------------------------------------------------$
$  COMMENTS  (OPTIONAL)                                                                                           $
$                                                                                                                 $
$    YOU MAY PUT NOTES HERE TO REMIND YOU IN THE FUTURE. THEY WILL BE                                             $
$  SHOWN IN THE OUTPUT FILE. THE NUMBER OF TOTAL ACTIVE COMMENT LINES                                             $
$  IS 50. THE ACTIVE WIDTH OF A LINE IS 72 COLUMNS.                                                               $
$------------------------------          INPUT YOUR COMMENTS HERE          ------------------------------$
This file describes a multilevel mine with mutiple atmospheric
junctions at different levels.  Several options are exercised in this
example.  A fire of 50,000 BTU/min heat output is simulated in airway
number 4.
$-----------------------------------------------------------------------------------------------------------------$
$  (2) CONTROL LINE I:                                                                                           $
$ NB,NFNUM,INFLOW,NVPN,NETW,TR,MADJ,ITN,NTEMP,TINC,SPAN,IOUT,TOUT,CONTC                                           $
$                                                                                                                 $
$       NB: NUMBER OF AIRWAYS IN NETWORK, MAXIMUM=500                                                             $
$       NFNUM: NUMBER OF FAN CHARACTERISTICS, MAXIMUM=10                                                          $
$       INFLOW: NUMBER OF CONTAMINATION SOURCES, MAXIMUM=10                                                       $
$       NVPN: MARKER FOR PRESENCE OF JUNCTION CARDS, 0=PRESENT                                                    $
$       NETW: MARKER, 1=TERMINATE AFTER NETWORK BALANCING                                                         $
$       TR: REFERENCE TEMPERATURE OF AIR, DEGREES F. DEFAULT VALUE=75                                             $
$       MADJ: MAX. NO. OF ITERATIONS IN TEMPERATURE PART. DEFAULT=10                                              $
$       ITN: MAX. NO. OF ITERATIONS IN DYNAMIC PART, DEFAULT=10                                                   $
$       NTEMP: MARKER, 1=TERMINATE AFTER STEADY STATE SIMULATION                                                  $
$       TINC: TIME INCREMENT IN DYNAMIC SIMULATION PART, SEC.                                                     $
$       SPAN: TIME SPAN OF DYNAMIC SIMULATION, MIN.                                                               $
$       IOUT: OUTPUT, 0: BRIEF; 1: NORMAL; -1: DETAIL; -2; MORE DETAILED                                          $
$       TOUT: TIME INTERVAL FOR OUTPUT, IN MIN.                                                                   $
$       CONCT: MARKER, 1=TRANSIENT SIMULATION FOR CONCENTRATION ONLY                                              $
```

```
$                              0=TRANSIENT SIMULATION                                   $
$                     ***  INPUT DATA FOR CONTROL LINE I HERE  ***                      $
$-NB,NFNUM,INFLOW,NVPN,NETW,TR,MADJ,ITN,NTEMP,TINC,SPAN,IOUT,TOUT,CONCT  $
  49    1        1       0       0       75.0   60  0     0        60.0    2      -2      0       0
$-----------------------------------------------------------------------------------------$
$  AIRWAY LINES                                                                          $
$     NB,JS,JF,NWTYP,R,Q,KF,LA,A,O,HA,HK,CH4V,CH4PA,TROCK                                $
$                                                                                        $
$  * IF NB IS 2 OR LARGER, THE NUMBER OF AIRWAY LINES MUST BE NB.                        $
$  IF NB IS 1 OR LESS, THE LAST AIRWAY LINE MUST BE 9999.                                $
$  * KF,LA,A,O,HA,HK HAVE THEIR AVERAGE OR DEFAULT VALUES WHICH ARE                      $
$  ACTIVATED BY INPUTING ZERO FOR THE RESPECTIVE TERMS.                                  $
$  * TROCK DOES NOT NEED TO INPUT EXCEPT TO OVERRIDE THE PROGRAM                         $
$  CALCULATED VALUE.                                                                     $
$  * COMMENT LINES ARE ACTIVE IN THIS DATA PARAGRAGH.                                    $
$                                                                                        $
$    NB:       AIRWAY ID NUMBER (1 TO 9980).                                             $
$    JS:       JUNCTION ID NUMBER OF AIRWAY BEGINNING (1 TO 999)                         $
$    JF:       JUNCTION ID NUMBER OF AIRWAY END (1 TO 999)                               $
$    NWTYP:  AIRWAY TYPE, 1=FAN; 0=ORDINARY; -1=FIXED-Q                                  $
$    R:        RESISTANCE OF AIRWAY 1*E-10 IN. W.G./(CFM)^2                              $
$    Q:        AIRFLOW RATE IN CFM                                                       $
$    KF:       FRICTION FACTOR IN 1*E-10 LBM*MIN2/FT^4                                   $
$    LA:       AIRWAY LENGTH IN FT                                                       $
$    A:        AIRWAY CROSS SECTIONAL AREA IN FT^2                                       $
$    O:        AIRWAY PERIMETER IN FT                                                    $
$    HA:       THERMAL DIFFUSIVITY OF ROCK, FT^2/HR                                      $
$    HK:       THERMAL CONDUCTIVITY OF ROCK, BTU/HR*FT*F                                 $
$    CH4V:     METHANE EMISION RATE IN AIRWAYS FT^3/MIN                                  $
$    CH4PA:    METHANE EMISION RATE PER UNIT SURFACE AREA FT/MIN                         $
$    TROCK:  AVERAGE ROCK TEMPERATURE IN AIRWAY DEGREES F                                $
$                                                                                        $
$                      ***  INPUT DATA FOR AIRWAY CARDS HERE  ***                        $
$--------------     NB,JS,JF,NWTYP,R,Q,KF,LA,A,O,HA,HK,CH4V,CH4PA,TROCK     --------------$
  1      5     32     0     0.0     9240     100      430     50.0      30.0,,,,,60
  2      5      6    -1     0.0    42000     100     1000     55.5      30.0,,,,,60
  3      8     30     0     0.0     3600     100      650     40.0      30.0,,,,,60
  4      6      8     0     0.0     5430     100      200     50.0      30.0,,,,,60
  5      6      7     0     0.0     5660     100       50     60.0      30.0,,,,,60
  6      7     31     0     0.0     4000     100       50     60.0      30.0,,,,,60
  7     30      9     0     0.0     3600     100      110     45.0      25.0,,,,,60
  8      8     10     0     0.0     1740     100      110     15.0      20.0,,,,,60
  9      7     12     0     0.0     1660     100      110     25.0      18.0,,,,,60
 10     31     14     0     0.0     4000     100      110     90.0      35.0,,,,,60
 11      9     28     0     0.0     6800     100      200     45.0      30.0,,,,,60
 12     10      9     0     0.0     3200     100      325     45.0     100.0,,,,,60
 13     11     10     0     0.0     1460     100      200     40.0      30.0,,,,,60
 14     12     11     0     0.0     1960     100      300     60.0      30.0,,,,,60
 15     13     12     0     0.0      400     100       50     55.0      30.0,,,,,60
 16     14     13     0     0.0     2800     100       80     35.0      22.0,,,,,60
 17     14     29     0     0.0     1000     100      350     35.0      30.0,,,,,60
 18     28     15     0     0.0     6800     100      150     80.0      50.0,,,,,60
 19     11     16     0     0.0      500     100      150     15.0      20.0,,,,,60
 20     13     19     0     0.0     2200     100      150     45.0      26.0,,,,,60
```

```
21      29      20      0       0.0     1000    100     150     5.0     20.0,,,,,60
22      16      15      0       0.0     1000    100     585     55.0    30.0,,,,,60
23      16      17      0       0.0     1700    100     55      28.0    20.0,,,,,60
24      18      17      0       0.0     750     100     90      35.0    25.0,,,,,60
25      19      18      0       0.0     0       100     385     30.0    25.0,,,,,60
26      19      20      0       0.0     900     100     650     40.0    24.0,,,,,60
27      15      22      0       0.0     5600    100     210     45.0    26.0,,,,,60
28      17      23      0       0.0     900     100     260     36.0    32.0,,,,,60
29      18      24      0       0.0     750     100     260     28.0    20.0,,,,,60
30      19      26      0       0.0     1000    100     260     7.0     20.0,,,,,60
31      20      26      0       0.0     1680    100     500     14.0    20.0,,,,,60
32      21      27      0       0.0     3120    100     255     30.0    22.0,,,,,60
33      22      21      0       0.0     4200    100     100     30.0    20.0,,,,,60
34      23      22      0       0.0     1400    100     585     30.0    20.0,,,,,60
35      23      24      0       0.0     2300    100     190     40.0    35.0,,,,,60
36      24      25      0       0.0     3000    100     95      40.0    30.0,,,,,60
37      25      26      0       0.0     1000    100     220     45.0    26.0,,,,,60
38      26      2       0       0.0     3000    100     450     35.0    25.0,,,,,60
39      21      1       0       0.0     1100    100     150     5.0     20.0,,,,,60
40      25      1       0       0.0     1750    100     150     8.0     20.0,,,,,60
41      33      27      0       0.0     100000  100     95      10000.0 400.0
42      27      34      0       0.0     100000  100     4       10000.0 400.0
43      34      3       0       0.0     100000  100     151     10000.0 400.0
44      4       3       0       0.0     100000  100     156     10000.0 400.0
45      3       5       1       5.0     1500    100     155     65.0    32.0,,,,,60
46      32      4       0       500.0   9500    100     10      30.0    25.0,,,,,60
47      23      1       0       0.0     500     100     150     5.0     20.0,,,,,60
48      1       33      0       0.0     10000   100     800     10000.0 400.0
49      2       34      0       5.0     3000    100     600     10000.0 400.0
$-------------------------------------------------------------------------------------------------------------------------$
$  JUNCTION LINES SECTION 1                                                                                              $
$     JNO,T,Z,CH4C                                                                                                        $
$                                                                                                                         $
$  * IF NVPN IS 0 OR LESS, EACH JUNCTION SPECIFIED BY JS AND JF  MUST                                                     $
$  HAVE  A ONE LINE INPUT HERE. OTHERWISE JUNCTION LINES MUST BE                                                          $
$  OMITTED.                                                                                                               $
$                                                                                                                         $
$        JNO:    JUNCTION ID NUMBER (1 TO 999)                                                                          $
$        T:      TEMPERATURE OF AIR IN JUNCTIONS DEGREES F                                                              $
$        Z:      ELEVATION OF JUNCTIONS  FT                                                                             $
$        CH4C:   METHANE CONCENTRATION IN JUNCTIONS PCT                                                                 $
$                                                                                                                         $
$                              ***   PLACE JUNCTION LINES HERE   ***                                                    $
$-------------------------------------------------  JNO,T,Z,CH4C  -------------------------------------------------$
   1        95.0        7690
   2        60.0        7591
   3        95.0        7440
   4        95.0        7284
   5        53.0        7286
   6        55.0        7289
   7        55.0        7289
   8        58.0        7289
   9        60.0        7365
  10        60.0        7365
```

```
 11        60.0        7365
 12        60.0        7365
 13        60.0        7365
 14        60.0        7365
 15        60.0        7470
 16        60.0        7470
 17        60.0        7470
 18        60.0        7470
 19        60.0        7470
 20        60.0        7470
 21        60.0        7595
 22        60.0        7595
 23        60.0        7595
 24        60.0        7594
 25        60.0        7593
 26        60.0        7593
 27        95.0        7595
 28        60.0        7365
 29        60.0        7365
 30        60.0        7290
 31        57.0        7289
 32        47.0        7284
 33        95.0        7690
 34        95.0        7591
$-----------------------------------------------------------------------------------------------------$
$  JUNCTION LINES SECTION 2 (JUNCTIONS IN THE ATMOSPHERE)                                            $
$  IF THERE IS  MORE THAN ONE JUNCTION IN THE ATMOSPHERE, THIS                                       $
$  SECTION IS EMPLOYED.                                                                              $
$                                                                                                    $
$     JAN :  THE NUMBER OF THE JUNCTIONS IN THE ATMOSPHERE                                           $
$     JA:    JUNCTION ID NUMBER                                                                      $
$     JATP:  JUNCTION TYPE:                                                                          $
$            0 = STARTING JUNCTION                                                                   $
$            1 = OTHER ATMOSPHERE JUNCTIONS EXCEPT STARTING JUNCTION                                 $
$                                                                                                    $
$             ***   INPUT DATA FOR JUNCTION LINES SECTION 2 HERE   ***                              $
$-------------------------------------------------- JAN ---------------------------------------------$
  6
$-------------------------------------------------- JA,JATP ------------------------------------------$
  1    1
  3    0
  4    1
 27    1
 33    1
 34    1
$-----------------------------------------------------------------------------------------------------$
$    FAN LINES 1 AND 2 COMPRISE A DATA GROUP WHICH MUST REPEAT NFNUM                                 $
$  TIMES FOR NFNUM FAN CURVES. IF NFNUM IS 0 OR LESS, FAN LINES 1 AND                                $
$  2 MUST BE OMITTED.                                                                                $
$                                                                                                    $
$  (6) FAN LINE 1                                                                                    $
$  NOF,MPTS                                                                                          $
$  NOF:     ID NUMBER OF FAN BRANCH.                                                                 $
$  MPTS:    NUMBER OF POINTS DEFINING FAN CHARACTERISTICS, MAXIMUM 10                                $
```

```
$                                                                                $
$  (7) FAN LINE 2  (IF MPTS IS LARGER THAN 5, CONTINUE ON 2ND LINE. )            $
$    QF,PF,QF,PF,QF,PF,QF,PF,QF,PF                                               $
$    QF:      AIRFLOW AT GIVEN POINT OF FAN CHAR, CFM.                           $
$    PF:      PRESSURE AT GIVEN POINT OF FAN CHAR, IN. W.G.                      $
$                                                                                $
$                        ***  INPUT FAN LINES 1 AND 2 HERE  ***                 $
$-------------------------------------------- NOF,MPTS ---------------------------------------------$
$------------------------------------- QF,PF,QF,PF,QF,PF,QF,PF,QF,PF ------------------------------------$
 45  10
 37500      2.00      41500      1.83      45000      1.64      48500      1.44      52500      1.17
 55000      0.98      57500      0.78      60000      0.58      62500      0.36      65000      0.15
$-------------------------------------------------------------------------------------------------------$
$    CONTAMINATION (FIRE SOURCE) LINES                                           $
$    NCENT,CONT,CONC,HEAT,O2MIN,SMPO2,HTPO2,QCENT,TPR                            $
$                                                                                $
$  * EACH FIRE SOURCE MUST HAVE  A ONE LINE INPUT HERE. THE NUMBER OF            $
$  TOTAL FIRE SOURCES WAS SPECIFIED BY "INFLOW" IN CONTROL LINE 1                $
$                                                                                $
$    NCENT: ID NUMBER OF AIRWAYS WITH A FIRE SOURCES                             $
$  FIXED HEAT OUTPUT FIRE                                                        $
$    CONT:    VOLUME FLOW RATE OF CONTAMINATED GAS INFLOW IN CFM                 $
$    CONC:    CONCENTRATION OF CONTAMINANT IN GAS INFLOW PCT                     $
$    HEAT:    HEAT ENTERING AIRWAY IN BTU/MIN                                    $
$  OXYGEN RICH FIRE                                                              $
$    O2MIN:   OXYGEN CONCENTRATION OF AIR CURRECT LEAVING FIRE ZONE PCT          $
$  FUEL RICH FIRE                                                                $
$    SMPO2:   CONTAMINANT PRODUCTION PER CFM OF OXYGEN DELIVERY                  $
$    HTPO2:   HEAT PRODUCTION (BTU) PER CFM OF OXYGEN DELIVERY                   $
$  FIRE DAMPING FACTOR                                                           $
$    QCENT:  REF. Q DEFINING FIRE CHARACTERISTICS CFM                            $
$    TPR:  LEADING TIME PERIOD IN MIN. FOR FIRE TO REACH FULL STRENGTH           $
$                                                                                $
$                        ***  INPUT DATA FOR FIRE SOURCES HERE  ***             $
$-------------------NCENT,CONT,CONC,HEAT,O2MIN,SMPO2,HTPO2,QCENT,TPR-------------------$
     4     200     100     50000
$-------------------------------------------------------------------------------------------------------$
$  CONTROL LINE 2                                                               $
$  NAV,JSTART,TSTART,TIME,CRITSM,CRITGS,CRITHT,WRNPR,WRNSM,WRNGS,WRNHT           $
$                                                                                $
$  * A SET OF DEFAULT VALUES ARE AVAILABLE. THEY CAN BE ACTIVATED BY             $
$  INPUTING ZERO FOR THE RESPECTIVE TERMS.                                       $
$                                                                                $
$    NAV: MARKER, 1 OR LARGER=AVERAGE VALUE LINE IS PRESENT                      $
$    JSTART: STARTING JUNCTION (ATMOSPHERE).                                     $
$    TSTART: TEMPERATURE AT THE STARTING JUNCTION, DEGREES F                     $
$    TIME: TIME SPAN TO ASSUME QUASI-EQUILIBRIUM, HOURS DEFAULT=5                $
$    CRITSM: ACCURACY IN FUME CALCULATIONS PCT DEFAULT=0.005                     $
$    CRITGS: ACCURACY IN METHAME CALCULATIONS, PCT DEFAULT=.01                   $
$    CRITHT: ACCURACY IN TEMPERATURE CALCULATIONS, F DEFAULT=0.1                 $
$    WRNPR: PRESSURE DROP WARNING CRITERIA, IN. W.G. DEFAULT=0.01                $
$    WRNGS: METHANE CONCENTRATION WARNING CRITERIA, PCT DEFAULT=1.0              $
$    WRNSM: FUME CONCENTRATION WARNING CRITERIA, PCT DEFAULT=0.05                $
$    WRNHT: HIGH TEMPERATURE WARNING CRITERIA, DEGREES F DEFAULT=100             $
```

```
$                                                                                $
$                    ***   INPUT DATA FOR CONTROL LINE 2 HERE   ***              $
$- NAV,JSTART,TSTART,TIME,CRITSM,CRITGS,CRITHT,WRNPR,WRNSM,WRNGS,WRNHT- $
 0    3    70.0  30.0   .00001   0.10  .20  .01  .00001   .05   95.0
$-------------------------------------------------------------------------------$
$  AVERAGE-VALUE LINE                                                           $
$     HAAVR,HKAVR,KFAVR,LAAVR,AAVR,OAVR                                         $
$                                                                                $
$  THIS LINE MUST EXIST IF NAV IN CONTROL LINE 2 IS 1 OR LARGER.                 $
$  THE DEFAULT VALUE CAN BE SELECTED BY INPUTING ZERO FOR THE                   $
$  RESPECTIVE TERM. WHEN NAV IS ZERO OR LESS, THE WHOLE SET OF                  $
$  DEFAULT VALUES WILL BE USED.                                                  $
$                                                                                $
$     HAAVR: AVERAGE THERMAL DIFFUSIVITY OF ROCK, FT^2/HR DEFAULT=0.1           $
$     HKAVR: AVERAGE THERMAL CONDUCTIVITY OF ROCK, BTU/HR*FT*F                  $
$              DEFAULT=3.0                                                       $
$     KFAVR: FRICTION FACTOR, 1*E-10 LBM*MIN^2/FT^4. DEFAULT=100                $
$     LAAVR: AVERAGE LENGTH OF AIRWAY FT DEFAULT=500                            $
$     AAVR: AVERAGE SECTIONAL AREA OF AIRWAY FT^2 DEFAULT=100                   $
$     OAVR: AVERAGE PERIMETER FT DEFAULT=40                                     $
$                                                                                $
$              ***   INPUT AVERAGE VALUE LINE HERE IF NEEDED   ***              $
$---------------------------- HAAVR,HKAVR,KFAVR,LAAVR,AAVR,OAVR ----------------------------$

$-------------------------------------------------------------------------------$
$     TIME TABLE FOR CONDITION CHANGES                                          $
$                                                                                $
$     THIS TIME TABLE CAN CONTAIN THE FOLLOWING CONDITION CHANGES. IT           $
$     MUST END WITH A ONE LINE INPUT OF A VALUE LARGER THAN "SPAN" ON           $
$     CONTROL LINE 1. THIS TIME TABLE MUST EXIST, BUT MAY CONTAIN ONLY          $
$     THE ENDING LINE. THE CONDITION CHANGE LINES MUST BE LISTED IN             $
$     ORDER FROM TIME ZERO. THESE LINES HAVE THE FORM:                          $
$              TS,ICODE,PARAMETERS                                               $
$     WHERE TS=TIME OF CONDITION CHANGE                                         $
$              ICODE=NATURE OF CONDITION CHANGE                                  $
$              PARAMETERS ARE DEFINED AND RELATED TO SPECIFIC ICODES             $
$                                                                                $
$     1) TS,1,NBR,R                                                             $
$        THIS INPUT CHANGES BRANCH "NBR" TO AN ORDINARY AIRWAY WITH             $
$        AIRFLOW RESISTANCE EQUAL TO "R" AT "TS" MIN AFTER TIME ZERO            $
$        NBR: AIRWAY ID NUMBER.                                                 $
$        R: RESISTANCE OF AIRWAY 1*E-10 IN. W.G./(CFM)2.                        $
$                                                                                $
$     2) TS,2,NBR,INDEX,QF1,PF1,QF2,PF2,QF3,PF3,QF4,PF4,QF5,PF5                 $
$        THIS INPUT CHANGES BRANCH "NBR" TO A FAN BRANCH WITH ITS               $
$        CHARACTERISTICS SPECIFIED BY "QF1" TO "PF5".                           $
$        INDEX=NUMBER OF DATA POINTS TO DEFINE FAN CHARACTERISTCS MAX=5$
$        QF=AIRFLOW RATE IN FT^3/MIN                                            $
$        PF=CORRESPONDING FAN PRESSURE IN IN. W.G.                              $
$                                                                                $
$     3) TS,3,NBR,CONT,CONC,HEAT,02MIN,SMPO2,HTPO2,QCENT,TPR                    $
$        THIS INPUT CHANGES BRANCH "NBR" TO A FIRE SOURCE BRANCH WITH           $
$        FIRE CHARACTERISTICS SPECIFIED BY "CONT" TO "TPR", WHICH               $
$        RETAIN THE SAME MEANING AS IN CONTAMINANT (FIRE) SOURCE LINE.          $
```

```
$                                                                              $
$     4) TS,4,NBR                                                              $
$        THIS INPUT CHANGES FIRE BRANCH "NBR" TO AN ORDINARY AIRWAY AT         $
$        TIME "TS".                                                            $
$                                                                              $
$     5) TS,5,TINC                                                             $
$        THIS INPUT CHANGES THE TIME INCREMENT IN THE DYNAMIC                  $
$        SIMULATION TO A NEW VALUE OF "TINC" IN SEC.                           $
$                                                                              $
$     6) TS,6,TOUT                                                             $
$        THIS INPUT CHANGES OUTPUT INTERVAL TO A NEW VALUE OF "TOUT"           $
$        IN MIN.                                                               $
$                                                                              $
$     7) TS,7,JDPP                                                             $
$        THIS ICODE FORCES DETAILED RECORDS TO BE SET UP FOR AIRWAYS           $
$        WHICH ARE ON THE IMMEDIATE DOWN STREAM SIDE OF JUNCTION "JDPP"        $
$        AT "TS" MIN. AFTER TIME ZERO.                                         $
$                                                                              $
$              ***   INPUT CONDITION CHANGE TIME TABLE HERE   ***             $
$--------------- (IT MUST END WITH A ONE-LINE VALUE LARGER THAN "SPAN") ---------$
              0.0,3,4,200,100,50000
              1100
```

Output file

****

OUTPUT OF PROGRAM MFIRE

****

This file describes a multilevel mine with mutiple atmospheric junctions at different levels. Several options are exercised in this example. A fire of 50,000 BTU/min heat output is simulated in airway number 4.

REF.TEMP.(TR): 75.0 DEG.F REF. DENSITY (DR): .0740 LBM/FT3

LIST OF UNITS USED IN THIS OUTPUT FILE:

LENGTH: FT.;	AREA: FT2;	VOLUME: FT3;	MASS: LBM;
AIRFLOW: FT3/MIN;	TEMP. F;	RESISTANCE:	IN.W.G./CFM2;
FRICTION FACTOR:	LBM*MIN2/FT4;	THERMAL DIFFUSIVITY:	FT2/HR;
THERMAL CONDUCTIVITY:	BTU/HR*FT*F;	CONCENTRATION:	PERCENTAGE;
HEADLOSS:	IN.W.G.;	HEAT INPUT:	BTU/MIN;
CH4 PRODUCTION:	FT3/MIN;		

* ATTENTION * A NEW BRANCH HAS BEEN SET UP FOR EACH FIRE SOURCE.

AVE.-VALUE CARD WAS OMITTED BY USER (NAV=0)

THE FOLLOWING AVERAGE VALUES WILL BE USED TO FILL ANY MISSING DATA:

THERMAL DIFFUSIVITY OF ROCK	.100	FT2/HR
ROCK THERMAL CONDUCTIVITY	3.000	BTU/(HR*FT*F)
FRICTION FACTOR	100	1.E-10 LBF*MIN2/FT4
AIRWAY LENGTH	500	FT
SECTIONAL AREA	100.000	FT2
PERIMETER	40.000	FT

NO NETWORK CALCULATION HAS BEEN PERFORMED, THESE ARE THE INPUT DATA

AIRWAY	FROM	TO	AIRFLOW	TYPE	LENGTH	AREA	R	K	PERI.
1	5	32	5528.	0	430.0	50.0	.196E+01	100	30.0
2	5	6	42000.	0	1000.0	55.5	.478E+01	100	30.0
3	8	30	10269.	0	650.0	40.0	.578E+01	100	30.0
4	6	35	14899.	0	.1	50.0	.455E-03	100	30.0
1050	35	8	14899.	0	200.0	50.0	.911E+00	100	30.0
5	6	7	27101.	0	50.0	60.0	.132E+00	100	30.0
6	7	31	16997.	0	50.0	60.0	.132E+00	100	30.0
7	30	9	10269.	0	110.0	45.0	.573E+00	100	25.0
8	8	10	4629.	0	110.0	15.0	.124E+02	100	20.0
9	7	12	10104.	0	110.0	25.0	.240E+01	100	18.0
10	31	14	16997.	0	110.0	90.0	.100E+00	100	35.0
11	9	28	17759.	0	200.0	45.0	.125E+01	100	30.0
12	10	9	7489.	0	325.0	45.0	.677E+01	100	00.0
13	11	10	2860.	0	200.0	40.0	.178E+01	100	30.0
14	12		10045.	0	300.0	60.0	.791E+00	100	30.0
15	13	12	-60.	0	50.0	55.0	.171E+00	100	30.0

16	14	13	16000.	0	80.0	35.0	.779E+00	100	22.0
17	14	29	997.	0	350.0	35.0	.465E+01	100	30.0
18	28	15	17759.	0	150.0	80.0	.278E+00	100	50.0
19	11	16	7185.	0	150.0	15.0	.169E+02	100	20.0
20	13	19	16059.	0	150.0	45.0	.812E+00	100	26.0
21	29	20	997.	0	150.0	5.0	.455E+03	100	20.0
22	16	15	696.	0	585.0	55.0	.200E+01	100	30.0
23	16	17	6489.	0	55.0	28.0	.951E+00	100	20.0
24	18	17	2458.	0	90.0	35.0	.996E+00	100	25.0
25	19	18	10712.	0	385.0	30.0	.677E+01	100	25.0
26	19	20	3234.	0	650.0	40.0	.463E+01	100	24.0
27	15	22	18454.	0	210.0	45.0	.114E+01	100	26.0
28	17	23	8947.	0	260.0	36.0	.338E+01	100	32.0
29	18	24	8254.	0	260.0	28.0	.450E+01	100	20.0
30	19	26	2113.	0	260.0	7.0	.288E+03	100	20.0
31	20	26	4231.	0	500.0	14.0	.692E+02	100	20.0
32	21	27	19957.	0	255.0	30.0	.394E+01	100	22.0
33	22	21	21537.	0	100.0	30.0	.141E+01	100	20.0
34	23	22	3083.	0	585.0	30.0	.822E+01	100	20.0
35	23	24	3846.	0	190.0	40.0	.197E+01	100	35.0
36	24	25	12100.	0	95.0	40.0	.845E+00	100	30.0
37	25	26	8179.	0	220.0	45.0	.119E+01	100	26.0
38	26	2	14523.	0	450.0	35.0	.498E+01	100	25.0
39	21	1	1580.	0	150.0	5.0	.455E+03	100	20.0
40	25	1	3921.	0	150.0	8.0	.111E+03	100	20.0
41	33	27	7520.	0	95.0	100.0	.721E-06	100	400.0
42	27	34	27477.	0	4.0	100.0	.304E-07	100	400.0
43	34	3	42000.	0	151.0	100.0	.115E-05	100	400.0
44	4	3	5528.	0	156.0	100.0	.118E-05	100	400.0
45	3	5	47528.	1	155.0	65.0	.144E+01	100	32.0
46	32	4	5528.	0	10.0	30.0	.500E+03	100	25.0
47	23	1	2018.	0	150.0	5.0	.455E+03	100	20.0
48	1	33	7520.	0	800.0	100.0	.607E-05	100	400.0
49	2	34	14523.	0	600.0	100.0	.500E+01	100	400.0

JUNCTION	TEMP.	ELEVATION	CH4	JUNCTION	TEMP.	ELEVATION	CH4
1	95.0	7690.0	.00	2	60.0	7591.0	.00
3	95.0	7440.0	.00	4	95.0	7284.0	.00
5	53.0	7286.0	.00	6	55.0	7289.0	.00
7	55.0	7289.0	.00	8	58.0	7289.0	.00
9	60.0	7365.0	.00	10	60.0	7365.0	.00
11	60.0	7365.0	.00	12	60.0	7365.0	.00
13	60.0	7365.0	.00	14	60.0	7365.0	.00
15	60.0	7470.0	.00	16	60.0	7470.0	.00
17	60.0	7470.0	.00	18	60.0	7470.0	.00
19	60.0	7470.0	.00	20	60.0	7470.0	.00
21	60.0	7595.0	.00	22	60.0	7595.0	.00
23	60.0	7595.0	.00	24	60.0	7594.0	.00
25	60.0	7593.0	.00	26	60.0	7593.0	.00
27	95.0	7595.0	.00	28	60.0	7365.0	.00
29	60.0	7365.0	.00	30	60.0	7290.0	.00

31	57.0	7289.0	.00	32	47.0	7284.0	.00
33	95.0	7690.0	.00	34	95.0	7591.0	.00
35	55.0	7289.0	.00				

NETWORK CALCULATION RESULTS

AIRWAY	FROM	TO	TYPE	RESISTANCE	AIRFLOW	HEADLOSS
1	5	32	0	.1959E+01	5528.	.006
2	5	6	0	.4778E+01	42000.	.843
3	8	30	0	.5783E+01	10269.	.061
4	6	35	0	.4555E-03	14899.	.000
1050	35	8	0	.9110E+00	14899.	.020
5	6	7	0	.1318E+00	27101.	.010
6	7	31	0	.1318E+00	16997.	.004
7	30	9	0	.5727E+00	10269.	.006
8	8	10	0	.1237E+02	4629.	.027
9	7	12	0	.2405E+01	10104.	.025
10	31	14	0	.1002E+00	16997.	.003
11	9	28	0	.1250E+01	17759.	.039
12	10	9	0	.6769E+01	7489.	.038
13	11	10	0	.1779E+01	2860.	.001
14	12	11	0	.7908E+00	10045.	.008
15	13	12	0	.1711E+00	-60.	.000
16	14	13	0	.7791E+00	16000.	.020
17	14	29	0	.4648E+01	997.	.000
18	28	15	0	.2780E+00	17759.	.009
19	11	16	0	.1687E+02	7185.	.087
20	13	19	0	.8123E+00	16059.	.021
21	29	20	0	.4555E+03	997.	.045
22	16	15	0	.2002E+01	696.	.000
23	16	17	0	.9510E+00	6489.	.004
24	18	17	0	.9960E+00	2458.	.001
25	19	18	0	.6766E+01	10712.	.078
26	19	20	0	.4626E+01	3234.	.005
27	15	22	0	.1137E+01	18454.	.039
28	17	23	0	.3384E+01	8947.	.027
29	18	24	0	.4496E+01	8254.	.031
30	19	26	0	.2877E+03	2113.	.128
31	20	26	0	.6916E+02	4231.	.124
32	21	27	0	.3943E+01	19957.	.157
33	22	21	0	.1406E+01	21537.	.065
34	23	22	0	.8224E+01	3083.	.008
35	23	24	0	.1972E+01	3846.	.003
36	24	25	0	.8451E+00	12100.	.012
37	25	26	0	.1191E+01	8179.	.008
38	26	2	0	.4980E+01	14523.	.105
39	21	1	0	.4555E+03	1580.	.114
40	25	1	0	.1112E+03	3921.	.171
41	33	27	0	.7212E-06	7520.	.000
42	27	34	0	.3037E-07	27477.	.000
43	34	3	0	.1146E-05	42000.	.000

44	4	3	0	.1184E-05	5528.	.000
46	32	4	0	.5000E+03	5528.	1.528
47	23	1	0	.4555E+03	2018.	.186
48	1	33	0	.6073E-05	7520.	.000
49	2	34	0	.5000E+01	14523.	.105

FANS

AIRWAY	FROM	TO	AIRFLOW	FAN PRESSURE
45	3	5	47528.	1.444

THESE CHARACTERISTICS WERE STORED FOR FANS 45

Q*1000	PF	Q*1000	PF	Q*1000	PF	Q*1000	PF	Q*1000	PF
37.5	2.00	41.5	1.83	45.0	1.64	48.5	1.44	52.5	1.17
55.0	.98	57.5	.78	60.0	.58	62.5	.36	65.0	.15

COEFF. OF POLYNOMIAL OF FAN CURVES
PF=A0+A1*Q+A2*Q**2+A3*Q**3

FAN 45

A0	A1	A2	A3	FROM QF	TO QF
.143E+02	-.907E-03	.231E-07	-.206E-12	.375E+05	.415E+05
-.204E+02	.160E-02	-.373E-07	.280E-12	.415E+05	.450E+05
.255E+02	-.146E-02	.307E-07	-.224E-12	.450E+05	485E+05
-.695E+01	.550E-03	-.107E-07	.608E-13	.485E+05	525E+05
.356E+01	-.509E-04	.724E-09	-.119E-13	.525E+05	.550E+05
-.544E+02	.311E-02	-.568E-07	.337E-12	.550E+05	.575E+05
.142E+03	-.712E-02	.121E-06	-.695E-12	.575E+05	.600E+05
-.260E+03	.129E-01	-.213E-06	.116E-11	.600E+05	.625E+05
-.344E+04	.163E+00	-.258E-05	.136E-10	.625E+05	.650E+05

INTERPOLATION:

Q*1000	PF	Q*1000	PF	Q*1000	PF	Q*1000	PF	Q*1000	PF
37.5	2.00	38.3	1.97	39.1	1.94	39.9	1.90	40.7	1.87
41.5	1.83	42.2	1.79	42.9	1.76	43.6	1.72	44.3	1.68
45.0	1.64	45.7	1.60	46.4	1.56	47.1	1.52	47.8	1.48
48.5	1.44	49.3	1.39	50.1	1.34	50.9	1.28	51.7	1.23
52.5	1.17	53.0	1.13	53.5	1.10	54.0	1.06	54.5	1.02

55.0	.98	55.5	.94	56.0	.90	56.5	.86	57.0	.82
57.5	.78	58.0	.74	58.5	.70	59.0	.66	59.5	.62
60.0	.58	60.5	.54	61.0	.49	61.5	.45	62.0	.40
62.5	.36	63.0	.31	63.5	.26	64.0	.20	64.5	.16

THE STATED NUMBER OF AIRWAYS WAS 50
THE STATED NUMBER OF JUNCTIONS WAS 35

DATA FOR CONCENTRATION AND TEMPERATURE CALCULATION

AIRWAY	FROM	TO	ELEV. DIFF.	ROCK T	CH4 PROD.	COND.	DIFFUSIVITY
1	5	32	-2.00	60.00	.0	3.00	.100
2	5	6	3.00	60.00	.0	3.00	.100
3	8	30	1.00	60.00	.0	3.00	.100
4	6	35	.00	60.00	.0	3.00	.100
1050	35	8	.00	60.00	.0	3.00	.100
5	6	7	.00	60.00	.0	3.00	.100
6	7	31	.00	60.00	.0	3.00	.100
7	30	9	75.00	60.00	.0	3.00	.100
8	8	10	76.00	60.00	.0	3.00	.100
9	7	12	76.00	60.00	.0	3.00	.100
10	31	14	76.00	60.00	.0	3.00	.100
11	9	28	.00	60.00	.0	3.00	.100
12	10	9	.00	60.00	.0	3.00	.100
13	11	10	.00	60.00	.0	3.00	.100
14	12	11	.00	60.00	.0	3.00	.100
15	13	12	.00	60.00	.0	3.00	.100
16	14	13	.00	60.00	.0	3.00	.100
17	14	29	.00	60.00	.0	3.00	.100
18	28	15	105.00	60.00	.0	3.00	.100
19	11	16	105.00	60.00	.0	3.00	.100
20	13	19	105.00	60.00	.0	3.00	.100
21	29	20	105.00	60.00	.0	3.00	.100
22	16	15	.00	60.00	.0	3.00	.100
23	16	17	.00	60.00	.0	3.00	.100
24	18	17	.00	60.00	.0	3.00	.100
25	19	18	.00	60.00	.0	3.00	.100
26	19	20	.00	60.00	.0	3.00	.100
27	15	22	125.00	60.00	.0	3.00	.100
28	17	23	125.00	60.00	.0	3.00	.100
29	18	24	124.00	60.00	.0	3.00	.100
30	19	26	123.00	60.00	.0	3.00	.100
31	20	26	123.00	60.00	.0	3.00	.100
32	21	27	.00	60.00	.0	3.00	.100
33	22	21	.00	60.00	.0	3.00	.100
34	23	22	.00	60.00	.0	3.00	.100
35	23	24	-1.00	60.00	.0	3.00	.100

36	24	25	-1.00	60.00	.0	3.00	.100
37	25	26	.00	60.00	.0	3.00	.100
38	26	2	-2.00	60.00	.0	3.00	.100
39	21	1	95.00	60.00	.0	3.00	.100
40	25	1	97.00	60.00	.0	3.00	.100
41	33	27	-95.00	94.69	.0	3.00	.100
42	27	34	-4.00	94.71	.0	3.00	.100
43	34	3	-151.00	94.53	.0	3.00	.100
44	4	3	156.00	95.43	.0	3.00	.100
45	3	5	-154.00	60.00	.0	3.00	.100
46	32	4	.00	60.00	.0	3.00	.100
47	23	1	95.00	60.00	.0	3.00	.100
48	1	33	.00	95.00	.0	3.00	.100
49	2	34	.00	95.00	.0	3.00	.100

A TEMPERATURE OF 70.0 WAS ASSIGNED TO JUNCTION NO. 3

THE FOLLOWING CONTAMINATION WAS ASSUMED

	FIXED INFLUX			O2-RICH	FUEL-RICH		LEADING T	REF. Q
AIRWAY	FLUX	%CONC	HEAT	%O2	FUMES	HEAT		
4	200.0	100.00	.500E+05	.00	.00	.0	.00	0

****

OUTPUT OF THE TEMPERATURE PART OF THE PROGRAM

****

THRESHOLD IN ACCURACY (SUM OF NVP CORRECTIONS PER MESH < 2.E-4 IN.W.G.)
SATISFIED. CURRENT SUMFNV PER MESH .000083 IN.W.G., ITERATIONS 2

TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN AIRWAYS

AIRWAY	FROM	TO	AIRFLOW	AVE. T	T AT END	FUMES	METHANE	HEADLOSS
1	5	32	5674.	65.80	63.05	00.00	00.00	.006
2	5	6	43031.	67.15	64.99	00.00	00.00	.859

3	8	30	10264.	61.90	60.83	00.00	00.00	.058
4	6	35	14980.	64.99	64.98	00.00	00.00	.000
1050	35	8	14980.	64.28	63.64	00.00	00.00	.020
5	6	7	28050.	64.87	64.76	00.00	00.00	.010
6	7	31	17312.	64.61	64.46	00.00	00.00	.004
7	30	9	10264.	60.57	60.32	00.00	00.00	.006
8	8	10	4716.	62.76	61.99	00.00	00.00	.026
9	7	12	10739.	64.19	63.65	00.00	00.00	.027
10	31	14	17312.	63.98	63.52	00.00	00.00	.003
11	9	28	17907.	60.17	60.15	00.00	00.00	.038
12	10	9	7643.	60.41	60.04	00.00	00.00	.037
13	11	10	2927.	61.43	60.90	00.00	00.00	.001
14	12	11	10205.	62.83	62.14	00.00	00.00	.008
15	12	13	534.	63.19	62.76	00.00	00.00	.000
16	14	13	16358.	63.35	63.19	00.00	00.00	.020
17	14	29	953.	61.27	60.28	00.00	00.00	.000
18	28	15	17907.	59.87	59.62	00.00	00.00	.008
19	11	16	7278.	61.48	60.90	00.00	00.00	.085
20	13	19	16892.	62.63	62.11	00.00	00.00	.022
21	29	20	953.	59.92	59.72	00.00	00.00	.039
22	16	15	1146.	60.31	60.06	00.00	00.00	.000
23	16	17	6132.	60.84	60.79	00.00	00.00	.003
24	18	17	2914.	60.82	60.69	00.00	00.00	.001
25	19	18	11236.	61.47	60.97	00.00	00.00	.081
26	19	20	3443.	60.91	60.28	00.00	00.00	.005
27	15	22	19053.	59.38	59.13	00.00	00.00	.039
28	17	23	9045.	60.26	59.87	00.00	00.00	.026
29	18	24	8322.	60.48	60.06	00.00	00.00	.029
30	19	26	2214.	60.49	59.77	00.00	00.00	.134
31	20	26	4396.	59.86	59.65	00.00	00.00	.126
32	21	27	21211.	59.41	59.49	00.00	00.00	.167
33	22	21	22561.	59.29	59.32	00.00	00.00	.067
34	23	22	3508.	59.94	59.98	00.00	00.00	.010
35	23	24	3634.	59.92	59.95	00.00	00.00	.002
36	24	25	11956.	60.02	60.02	00.00	00.00	.011
37	25	26	8279.	60.02	60.02	00.00	00.00	.008
38	26	2	14889.	59.91	59.94	00.00	00.00	.104
39	21	1	1350.	59.63	59.70	00.00	00.00	.078
40	25	1	3677.	59.82	59.66	00.00	00.00	.142
41	33	27	6931.	94.92	94.96	00.00	00.00	.000
42	27	34	28141.	95.00	95.00	00.00	00.00	.000
43	34	3	43031.	94.97	95.02	00.00	00.00	.000
44	4	3	5674.	95.14	95.01	00.00	00.00	.000
45	3	5	48704.	69.93	69.86	00.00	00.00	1.441
46	32	4	5674.	63.00	62.96	00.00	00.00	1.538
47	23	1	1903.	59.79	59.71	00.00	00.00	.156
48	1	33	6931.	95.00	95.00	00.00	00.00	.000
49	2	34	14889.	92.79	95.00	00.00	00.00	.118

DATA FOR FIRE SOURCES

AIRWAY	AIRFLOW	TEMPERATURE	FUME	O2 LEFT	HEAT INPUT
4	14980.	64.99	00.00	.00	.000E+00

PARAMETERS OF AIR IN JUNCTIONS

JUNCTION	TEMP.	FUMES	METHANE	JUNCTION	TEMP.	FUMES	METHANE
1	95.00	00.00	00.00	2	59.94	00.00	00.00
3	70.00	00.00	00.00	4	95.00	00.00	00.00
5	69.86	00.00	00.00	6	64.99	00.00	00.00
7	64.76	00.00	00.00	8	63.64	00.00	00.00
9	60.20	00.00	00.00	10	61.57	00.00	00.00
11	62.14	00.00	00.00	12	63.65	00.00	00.00
13	63.18	00.00	00.00	14	63.52	00.00	00.00
15	59.65	00.00	00.00	16	60.90	00.00	00.00
17	60.76	00.00	00.00	18	60.97	00.00	00.00
19	62.11	00.00	00.00	20	60.16	00.00	00.00
21	59.32	00.00	00.00	22	59.26	00.00	00.00
23	59.87	00.00	00.00	24	60.02	00.00	00.00
25	60.02	00.00	00.00	26	59.87	00.00	00.00
27	95.00	00.00	00.00	28	60.15	00.00	00.00
29	60.28	00.00	00.00	30	60.83	00.00	00.00
31	64.46	00.00	00.00	32	63.05	00.00	00.00
33	95.00	00.00	00.00	34	95.00	00.00	00.00
35	64.98	00.00	00.00				

REVERSAL OF AIRFLOW HAS OCCURRED IN THE FOLLOWING PLACES

AIRWAY 15 IS NOW CARRYING AIR FROM 12 TO 13

IN THE FOLLOWING AIRWAYS EXIST CRITICAL CONDITIONS

AIRWAY	FROM	TO	CH4 % > .050	FUMES % > .000	TEMPERATURE > 95. F	HEADLOSS < .010 IN.WG.
1	5	32	00.00	00.000	63.0	.006
4	6	35	00.00	00.000	65.0	.000
5	6	7	00.00	00.000	64.8	.010
6	7	31	00.00	00.000	64.5	.004
7	30	9	00.00	00.000	60.3	.006
10	31	14	00.00	00.000	63.5	.003
13	11	10	00.00	00.000	60.9	.001
14	12	11	00.00	00.000	62.1	.008
15	12	13	00.00	00.000	62.8	.000
17	14	29	00.00	00.000	60.3	.000
18	28	15	00.00	00.000	59.6	.008
22	16	15	00.00	00.000	60.1	.000
23	16	17	00.00	00.000	60.8	.003

24	18	17	00.00	00.000	60.7	.001
26	19	20	00.00	00.000	60.3	.005
34	23	22	00.00	00.000	60.0	.010
35	23	24	00.00	00.000	60.0	.002
37	25	26	00.00	00.000	60.0	.008
41	33	27	00.00	00.000	95.0	.000
42	27	34	00.00	00.000	95.0	.000
43	34	3	00.00	00.000	95.0	.000
44	4	3	00.00	00.000	95.0	.000
48	1	33	00.00	00.000	95.0	.000
49	2	34	00.00	00.000	95.0	.118

IN THE FOLLOWING JUNCTIONS EXIST CRITICAL CONDITIONS

JUNCTION	CH4 % > .05	FUMES % > .000	TEMP. F > 95.	JUNCTION	CH4 % > .05	FUMES % > .000	TEMP. F > 95.
1	00.00	00.00	95.0	4	00.00	00.00	95.0
27	00.00	00.00	95.0	33	00.00	00.00	95.0
34	00.00	00.00	95.0				

****

NON-STEADY STATE SIMULATION OUTPUT

****

* ATTENTION * BRANCH 4 CAUGHT IN FIRE AT .00 MIN.
LEADING TIME: .0 PARAMETERS AS FOLLOWS:

FIXED-HEAT-INPUT FLOWRATE	CONC.	HEAT	OXYGEN-RICH (%)	FUEL-RICH FUME P O2	HEAT PO2	REF. Q
200.	100.00	.500E+05	.00	.00	.00	0.

TIME AT 60. SEC. AFTER EVENT

THRESHOLD IN ACCURACY (SUM OF NVP CORRECTIONS PER MESH < 2.E-4 IN.W.G.)
SATISFIED. CURRENT SUMFNV PER MESH .000096 IN.W.G., ITERATIONS 3

TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN AIRWAYS

AIRWAY	FROM	TO	DELTA Q	AIRFLOW	AVE. T	T AT END	FUMES	CH4	HEADLOSS
1	5	32	1.	5675.	65.80	63.05	00.00	00.00	.006
2	5	6	-7.	43024.	67.15	64.99	00.00	00.00	.859
3	8	30	-2506.	7758.	83.90	60.83	00.00	00.00	.036
4	6	35	2939.	17919.	234.04	234.04	1.19	00.00	.000
1050	35	8	2939.	17919.	209.40	187.20	1.19	00.00	.046
5	6	7	-2946.	25105.	64.87	64.76	00.00	00.00	.008
6	7	31	-1792.	15520.	64.61	64.46	00.00	00.00	.003
7	30	9	-2506.	7758.	60.57	60.32	00.00	00.00	.003
8	8	10	5445.	10161.	162.25	140.80	1.19	00.00	.173
9	7	12	-1154.	9585.	64.19	63.65	00.00	00.00	.021
10	31	14	-1792.	15520.	63.98	63.52	00.00	00.00	.002
11	9	28	-1309.	16598.	60.17	60.15	00.00	00.00	.033
12	10	9	1197.	8840.	75.55	60.04	00.00	00.00	.053
13	10	11	4248.	1321.	61.43	62.14	00.00	00.00	.000
14	12	11	-3517.	6688.	62.83	62.14	00.00	00.00	.003
15	12	13	2363.	2897.	63.19	62.76	00.00	00.00	.000
16	14	13	-1766.	14593.	63.35	63.19	00.00	00.00	.016
17	14	29	-26.	927.	61.27	60.28	00.00	00.00	.000
18	28	15	-1309.	16598.	59.87	59.62	00.00	00.00	.007
19	11	16	731.	8009.	61.48	60.90	00.00	00.00	.103
20	13	19	597.	17490.	62.63	62.11	00.00	00.00	.024
21	29	20	-26.	927.	59.92	59.72	00.00	00.00	.037
22	16	15	1158.	2304.	60.31	60.06	00.00	00.00	.001
23	16	17	-426.	5706.	60.84	60.79	00.00	00.00	.003
24	18	17	421.	3335.	60.82	60.69	00.00	00.00	.001
25	19	18	429.	11664.	61.47	60.97	00.00	00.00	.087
26	19	20	120.	3563.	60.91	60.28	00.00	00.00	.006
27	15	22	-152.	18901.	59.38	59.13	00.00	00.00	.038
28	17	23	-5.	9040.	60.26	59.87	00.00	00.00	.026
29	18	24	8.	8330.	60.48	60.06	00.00	00.00	.030
30	19	26	48.	2263.	60.49	59.77	00.00	00.00	.139
31	20	26	94.	4490.	59.86	59.65	00.00	00.00	.132
32	21	27	-29.	21182.	59.41	59.49	00.00	00.00	.167
33	22	21	-33.	22528.	59.29	59.32	00.00	00.00	.067
34	23	22	119.	3627.	59.94	59.98	00.00	00.00	.010
35	23	24	-124.	3510.	59.92	59.95	00.00	00.00	.002
36	24	25	-116.	11840.	60.02	60.02	00.00	00.00	.011
37	25	26	-121.	8157.	60.02	60.02	00.00	00.00	.007
38	26	2	21.	14910.	59.91	59.94	00.00	00.00	.105
39	21	1	-4.	1346.	59.63	59.70	00.00	00.00	.078
40	25	1	5.	3682.	59.82	59.66	00.00	00.00	.142
41	33	27	1.	6932.	94.92	94.96	00.00	00.00	.000
42	27	34	-28.	28114.	95.00	95.00	00.00	00.00	.000

43	34	3	-7.	43024.	94.97	95.02	00.00	00.00	.000
44	4	3	1.	5675.	95.14	95.01	00.00	00.00	.000
45	3	5	-6.	48698.	69.93	69.86	00.00	00.00	1.441
46	32	4	1.	5675.	63.00	62.96	00.00	00.00	1.539
47	23	1	0.	1903.	59.79	59.71	00.00	00.00	.156
48	1	33	1.	6932.	95.00	95.00	00.00	00.00	.000
49	2	34	21.	14910.	92.79	95.00	00.00	00.00	.119

DATA FOR THE FUME FRONT IN AIRWAYS

AIRWAY	POSITION	FROM	TEMPERATURE	FUME	METHANE
3	131.09	8	158.19	1.19	00.00
12	121.12	10	80.01	1.10	00.00

DATA FOR FIRE SOURCES

AIRWAY	AIRFLOW	TEMPERATURE	FUME	O2 LEFT	HEAT INPUT
4	17919.	234.04	1.19	19.79	.498E+05

PARAMETERS OF AIR IN JUNCTIONS

JUNCTION	TEMP.	FUMES	METHANE	JUNCTION	TEMP.	FUMES	METHANE
1	95.00	00.00	00.00	2	59.94	00.00	00.00
3	70.00	00.00	00.00	4	95.00	00.00	00.00
5	69.86	00.00	00.00	6	64.99	00.00	00.00
7	64.76	00.00	00.00	8	187.20	1.19	00.00
9	60.18	00.00	00.00	10	134.51	1.10	00.00
11	62.14	00.00	00.00	12	63.65	00.00	00.00
13	63.15	00.00	00.00	14	63.52	00.00	00.00
15	59.66	00.00	00.00	16	60.90	00.00	00.00
17	60.75	00.00	00.00	18	60.97	00.00	00.00
19	62.11	00.00	00.00	20	60.16	00.00	00.00
21	59.32	00.00	00.00	22	59.26	00.00	00.00
23	59.87	00.00	00.00	24	60.02	00.00	00.00
25	60.02	00.00	00.00	26	59.87	00.00	00.00
27	95.00	00.00	00.00	28	60.15	00.00	00.00
29	60.28	00.00	00.00	30	60.83	00.00	00.00
31	64.46	00.00	00.00	32	63.05	00.00	00.00
33	95.00	00.00	00.00	34	95.00	00.00	00.00
35	234.04	1.19	00.00				

REVERSAL OF AIRFLOW HAS OCCURRED IN THE FOLLOWING PLACES

AIRWAY 13 IS NOW CARRYING AIR FROM 10 TO 11
AIRWAY 15 IS NOW CARRYING AIR FROM 12 TO 13

DATA RECORD IN AIRWAYS

AIRWAY	DIST.	FROM	FRONT T	REAR T	FUME	CH4	AIR M (LB)
3	131.09	8	158.19	187.20	1.19	00.00	331.69
1050	NO C.V. T(BGG)= 234.0, T(END)= 187.2,					FUME= 1.19%.	
8	NO C.V. T(BGG)= 187.2, T(END)= 140.8,					FUME= 1.19%.	
12	121.12	10	80.01	134.51	1.10	00.00	385.12

IN THE FOLLOWING AIRWAYS EXIST CRITICAL CONDITIONS

AIRWAY	FROM	TO	CH4 % > .050	FUMES % > .000	TEMPERATURE > 95. F	HEADLOSS < .010 IN.WG.
1	5	32	00.00	00.000	63.0	.006
4	6	35	00.00	1.193	234.0	.000
1050	35	8	00.00	1.193	187.2	.046
5	6	7	00.00	00.000	64.8	.008
6	7	31	00.00	00.000	64.5	.003
7	30	9	00.00	00.000	60.3	.003
8	8	10	00.00	1.193	140.8	.173
10	31	14	00.00	00.000	63.5	.002
13	10	11	00.00	00.000	62.1	.000
14	12	11	00.00	00.000	62.1	.003
15	12	13	00.00	00.000	62.8	.000
17	14	29	00.00	00.000	60.3	.000
18	28	15	00.00	00.000	59.6	.007
22	16	15	00.00	00.000	60.1	.001
23	16	17	00.00	00.000	60.8	.003
24	18	17	00.00	00.000	60.7	.001
26	19	20	00.00	00.000	60.3	.006
35	23	24	00.00	00.000	60.0	.002
37	25	26	00.00	00.000	60.0	.007
41	33	27	00.00	00.000	95.0	.000
42	27	34	00.00	00.000	95.0	.000
43	34	3	00.00	00.000	95.0	.000
44	4	3	00.00	00.000	95.0	.000
48	1	33	00.00	00.000	95.0	.000
49	2	34	00.00	00.000	95.0	.119

IN THE FOLLOWING JUNCTIONS EXIST CRITICAL CONDITIONS

JUNCTION	CH4 % > .05	FUMES % > .000	TEMP. F > 95.	JUNCTION	CH4 % > .05	FUMES % > .000	TEMP. F > 95.
1	00.00	00.00	95.0	4	00.00	00.00	95.0
8	00.00	1.19	187.2	10	00.00	1.10	134.5
27	00.00	00.00	95.0	33	00.00	00.00	95.0
34	00.00	00.00	95.0	35	00.00	1.19	234.0

****

TIME AT 120. SEC. AFTER EVENT

THRESHOLD IN ACCURACY (SUM OF NVP CORRECTIONS PER MESH < 2.E-4 IN.W.G.)
SATISFIED. CURRENT SUMFNV PER MESH .000170 IN.W.G., ITERATIONS 2

TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN AIRWAYS

AIRWAY	FROM	TO	DELTA Q	AIRFLOW	AVE. T	T AT END	FUMES	CH4	HEADLOSS
1	5	32	0.	5675.	65.80	63.05	00.00	00.00	.006
2	5	6	-2.	43022.	67.15	64.99	00.00	00.00	.859
3	8	30	583.	8340.	86.98	60.83	00.00	00.00	.042
4	6	35	-919.	17000.	225.23	225.23	1.13	00.00	.000
1050	35	8	-919.	17000.	82.48	147.83	1.13	00.00	.038
5	6	7	917.	26021.	64.87	64.76	00.00	00.00	.009
6	7	31	588.	16108.	64.61	64.46	00.00	00.00	.003
7	30	9	583.	8340.	60.57	60.32	00.00	00.00	.004
8	8	10	-1501.	8660.	121.20	100.58	1.13	00.00	.110
9	7	12	329.	9914.	64.19	63.65	00.00	00.00	.023
10	31	14	588.	16108.	63.98	63.52	00.00	00.00	.002
11	9	28	293.	16891.	60.17	60.15	00.00	00.00	.034
12	10	9	-289.	8551.	67.56	60.04	00.00	00.00	.048
13	10	11	-1212.	109.	64.43	62.14	00.00	00.00	.000
14	12	11	1068.	7756.	62.83	62.14	00.00	00.00	.005
15	12	13	-740.	2158.	63.19	62.76	00.00	00.00	.000
16	14	13	578.	15171.	63.35	63.19	00.00	00.00	.017
17	14	29	10.	937.	61.27	60.28	00.00	00.00	.000

18	28	15	293.	16891.	59.87	59.62	00.00	00.00	.007
19	11	16	-144.	7865.	61.48	60.90	00.00	00.00	.099
20	13	19	-162.	17328.	62.63	62.11	00.00	00.00	.023
21	29	20	10.	937.	59.92	59.72	00.00	00.00	.038
22	16	15	-251.	2052.	60.31	60.06	00.00	00.00	.001
23	16	17	107.	5813.	60.84	60.79	00.00	00.00	.003
24	18	17	-106.	3228.	60.82	60.69	00.00	00.00	.001
25	19	18	-116.	11549.	61.47	60.97	00.00	00.00	.086
26	19	20	-33.	3530.	60.91	60.28	00.00	00.00	.005
27	15	22	42.	18943.	59.38	59.13	00.00	00.00	.038
28	17	23	1.	9041.	60.26	59.87	00.00	00.00	.026
29	18	24	-9.	8321.	60.48	60.06	00.00	00.00	.029
30	19	26	-13.	2249.	60.49	59.77	00.00	00.00	.138
31	20	26	-22.	4467.	59.86	59.65	00.00	00.00	.130
32	21	27	6.	21188.	59.41	59.49	00.00	00.00	.167
33	22	21	7.	22536.	59.29	59.32	00.00	00.00	.067
34	23	22	-35.	3592.	59.94	59.98	00.00	00.00	.010
35	23	24	36.	3546.	59.92	59.95	00.00	00.00	.002
36	24	25	27.	11866.	60.02	60.02	00.00	00.00	.011
37	25	26	29.	8186.	60.02	60.02	00.00	00.00	.008
38	26	2	-7.	14903.	59.91	59.94	00.00	00.00	.104
39	21	1	1.	1347.	59.63	59.70	00.00	00.00	.078
40	25	1	-2.	3680.	59.82	59.66	00.00	00.00	.142
41	33	27	-2.	6930.	94.92	94.96	00.00	00.00	.000
42	27	34	5.	28119.	95.00	95.00	00.00	00.00	.000
43	34	3	-2.	43022.	94.97	95.02	00.00	00.00	.000
44	4	3	0.	5675.	95.14	95.01	00.00	00.00	.000
45	3	5	-2.	48697.	69.93	69.86	00.00	00.00	1.441
46	32	4	0.	5675.	63.00	62.96	00.00	00.00	1.539
47	23	1	-0.	1903.	59.79	59.71	00.00	00.00	.156
48	1	33	-2.	6930.	95.00	95.00	00.00	00.00	.000
49	2	34	-7.	14903.	92.79	95.00	00.00	00.00	.119

DATA FOR THE FUME FRONT IN AIRWAYS

AIRWAY	POSITION	FROM	TEMPERATURE	FUME	METHANE
3	329.09	8	98.06	1.19	00.00
12	303.57	10	60.77	1.10	00.00
13	16.04	10	97.98	1.13	00.00

DATA FOR FIRE SOURCES

AIRWAY	AIRFLOW	TEMPERATURE	FUME	O2 LEFT	HEAT INPUT
4	17000.	225.23	1.13	19.86	.498E+05

PARAMETERS OF AIR IN JUNCTIONS

JUNCTION	TEMP.	FUMES	METHANE	JUNCTION	TEMP.	FUMES	METHANE
1	95.00	00.00	00.00	2	59.94	00.00	00.00
3	70.00	00.00	00.00	4	95.00	00.00	00.00
5	69.86	00.00	00.00	6	64.99	00.00	00.00
7	64.76	00.00	00.00	8	147.83	1.13	00.00
9	60.17	00.00	00.00	10	100.58	1.13	00.00
11	62.14	00.00	00.00	12	63.65	00.00	00.00
13	63.13	00.00	00.00	14	63.52	00.00	00.00
15	59.67	00.00	00.00	16	60.90	00.00	00.00
17	60.75	00.00	00.00	18	60.97	00.00	00.00
19	62.11	00.00	00.00	20	60.16	00.00	00.00
21	59.32	00.00	00.00	22	59.26	00.00	00.00
23	59.87	00.00	00.00	24	60.02	00.00	00.00
25	60.02	00.00	00.00	26	59.87	00.00	00.00
27	95.00	00.00	00.00	28	60.15	00.00	00.00
29	60.28	00.00	00.00	30	60.83	00.00	00.00
31	64.46	00.00	00.00	32	63.05	00.00	00.00
33	95.00	00.00	00.00	34	95.00	00.00	00.00
35	225.23	1.13	00.00				

REVERSAL OF AIRFLOW HAS OCCURRED IN THE FOLLOWING PLACES

AIRWAY 13 IS NOW CARRYING AIR FROM 10 TO 11
AIRWAY 15 IS NOW CARRYING AIR FROM 12 TO 13

DATA RECORD IN AIRWAYS

AIRWAY	DIST.	FROM	FRONT T	REAR T	FUME	CH4	AIR M (LB)
3	110.91	8	114.59	147.83	1.13	00.00	299.23
	329.09	8	98.06	109.31	1.19	00.00	628.71
1050	NO C.V. T(BGG)= 225.2, T(END)= 147.8,					FUME= 1.13%.	
8	NO C.V. T(BGG)= 147.8, T(END)= 100.6,					FUME= 1.13%.	
12	158.99	10	62.72	100.58	1.13	00.00	532.05
	303.57	10	60.77	62.85	1.10	00.00	495.75
13	16.04	10	97.98	100.58	1.13	00.00	45.84

IN THE FOLLOWING AIRWAYS EXIST CRITICAL CONDITIONS

AIRWAY	FROM	TO	CH4 % > .050	FUMES % > .000	TEMPERATURE > 95. F	HEADLOSS < .010 IN.WG.
1	5	32	00.00	00.000	63.0	.006
4	6	35	00.00	1.131	225.2	.000
1050	35	8	00.00	1.131	147.8	.038
5	6	7	00.00	00.000	64.8	.009
6	7	31	00.00	00.000	64.5	.003
7	30	9	00.00	00.000	60.3	.004
8	8	10	00.00	1.131	100.6	.110
10	31	14	00.00	00.000	63.5	.002
13	10	11	00.00	00.000	62.1	.000
14	12	11	00.00	00.000	62.1	.005
15	12	13	00.00	00.000	62.8	.000
17	14	29	00.00	00.000	60.3	.000
18	28	15	00.00	00.000	59.6	.007
22	16	15	00.00	00.000	60.1	.001
23	16	17	00.00	00.000	60.8	.003
24	18	17	00.00	00.000	60.7	.001
26	19	20	00.00	00.000	60.3	.005
35	23	24	00.00	00.000	60.0	.002
37	25	26	00.00	00.000	60.0	.008
41	33	27	00.00	00.000	95.0	.000
42	27	34	00.00	00.000	95.0	.000
43	34	3	00.00	00.000	95.0	.000
44	4	3	00.00	00.000	95.0	.000
48	1	33	00.00	00.000	95.0	.000
49	2	34	00.00	00.000	95.0	.119

IN THE FOLLOWING JUNCTIONS EXIST CRITICAL CONDITIONS

JUNCTION	CH4 % > .05	FUMES % > .000	TEMP. F > 95.	JUNCTION	CH4 % > .05	FUMES % > .000	TEMP. F > 95.
1	00.00	00.00	95.0	4	00.00	00.00	95.0
8	00.00	1.13	147.8	10	00.00	1.13	100.6
27	00.00	00.00	95.0	33	00.00	00.00	95.0
34	00.00	00.00	95.0	35	00.00	1.13	225.2

****

OUTPUT OF THE QUASI-EQUILIBRIUM SIMULATION PART

****

REF.TEMP.(TR): 75.0 DEG.F REF. DENSITY (DR): .0740 LBM/FT3

LIST OF UNITS USED IN THIS OUTPUT FILE:

LENGTH: FT.;	AREA: FT2;	VOLUME: FT3;	MASS: LBM;
AIRFLOW: FT3/MIN;	TEMP. F;	RESISTANCE:	IN.W.G./CFM2;
FRICTION FACTOR:	LBM*MIN2/FT4;	THERMAL DIFFUSIVITY:	FT2/HR;
THERMAL CONDUCTIVITY:	BTU/HR*FT*F;	CONCENTRATION:	PERCENTAGE;
HEADLOSS:	IN.W.G.;	HEAT INPUT:	BTU/MIN;
CH4 PRODUCTION:	FT3/MIN;		

NO NETWORK CALCULATION HAS BEEN PERFORMED, THESE ARE THE INPUT DATA

AIRWAY	FROM	TO	AIRFLOW	TYPE	LENGTH	AREA	R	K	PERI.
1	5	32	5528.	0	430.0	50.0	.196E+01	100	30.0
2	5	6	42000.	0	1000.0	55.5	.478E+01	100	30.0
3	8	30	10269.	0	650.0	40.0	.578E+01	100	30.0
4	6	35	14899.	0	.1	50.0	.455E-03	100	30.0
1050	35	8	14899.	0	200.0	50.0	.911E+00	100	30.0
5	6	7	27101.	0	50.0	60.0	.132E+00	100	30.0
6	7	31	16997.	0	50.0	60.0	.132E+00	100	30.0
7	30	9	10269.	0	110.0	45.0	.573E+00	100	25.0
8	8	10	4629.	0	110.0	15.0	.124E+02	100	20.0
9	7	12	10104.	0	110.0	25.0	.240E+01	100	18.0
10	31	14	16997.	0	110.0	90.0	.100E+00	100	35.0
11	9	28	17759.	0	200.0	45.0	.125E+01	100	30.0
12	10	9	7489.	0	325.0	45.0	.677E+01	100	100.0
13	11	10	2860.	0	200.0	40.0	.178E+01	100	30.0
14	12	11	10045.	0	300.0	60.0	.791E+00	100	30.0
15	13	12	-60.	0	50.0	55.0	.171E+00	100	30.0
16	14	13	16000.	0	80.0	35.0	.779E+00	100	22.0
17	14	29	997.	0	350.0	35.0	.465E+01	100	30.0
18	28	15	17759.	0	150.0	80.0	.278E+00	100	50.0
19	11	16	7185.	0	150.0	15.0	.169E+02	100	20.0
20	13	19	16059.	0	150.0	45.0	.812E+00	100	26.0
21	29	20	997.	0	150.0	5.0	.455E+03	100	20.0
22	16	15	696.	0	585.0	55.0	.200E+01	100	30.0
23	16	17	6489.	0	55.0	28.0	.951E+00	100	20.0
24	18	17	2458.	0	90.0	35.0	.996E+00	100	25.0
25	19	18	10712.	0	385.0	30.0	.677E+01	100	25.0
26	19	20	3234.	0	650.0	40.0	.463E+01	100	24.0
27	15	22	18454.	0	210.0	45.0	.114E+01	100	26.0
28	17	23	8947.	0	260.0	36.0	.338E+01	100	32.0
29	18	24	8254.	0	260.0	28.0	.450E+01	100	20.0
30	19	26	2113.	0	260.0	7.0	.288E+03	100	20.0

31	20	26	4231.	0	500.0	14.0	.692E+02	100	20.0
32	21	27	19957.	0	255.0	30.0	.394E+01	100	22.0
33	22	21	21537.	0	100.0	30.0	.141E+01	100	20.0
34	23	22	3083.	0	585.0	30.0	.822E+01	100	20.0
35	23	24	3846.	0	190.0	40.0	.197E+01	100	35.0
36	24	25	12100.	0	95.0	40.0	.845E+00	100	30.0
37	25	26	8179.	0	220.0	45.0	.119E+01	100	26.0
38	26	2	14523.	0	450.0	35.0	.498E+01	100	25.0
39	21	1	1580.	0	150.0	5.0	.455E+03	100	20.0
40	25	1	3921.	0	150.0	8.0	.111E+03	100	20.0
41	33	27	7520.	0	95.0	100.0	.721E-06	100	400.0
42	27	34	27477.	0	4.0	100.0	.304E-07	100	400.0
43	34	3	42000.	0	151.0	100.0	.115E-05	100	400.0
44	4	3	5528.	0	156.0	100.0	.118E-05	100	400.0
45	3	5	47528.	1	155.0	65.0	.144E+01	100	32.0
46	32	4	5528.	0	10.0	30.0	.500E+03	100	25.0
47	23	1	2018.	0	150.0	5.0	.455E+03	100	20.0
48	1	33	7520.	0	800.0	100.0	.607E-05	100	400.0
49	2	34	14523.	0	600.0	100.0	.500E+01	100	400.0

JUNCTION	TEMP.	ELEVATION	CH4	JUNCTION	TEMP.	ELEVATION	CH4
1	95.0	7690.0	.00	2	60.0	7591.0	.00
3	95.0	7440.0	.00	4	95.0	7284.0	.00
5	53.0	7286.0	.00	6	55.0	7289.0	.00
7	55.0	7289.0	.00	8	58.0	7289.0	.00
9	60.0	7365.0	.00	10	60.0	7365.0	.00
11	60.0	7365.0	.00	12	60.0	7365.0	.00
13	60.0	7365.0	.00	14	60.0	7365.0	.00
15	60.0	7470.0	.00	16	60.0	7470.0	.00
17	60.0	7470.0	.00	18	60.0	7470.0	.00
19	60.0	7470.0	.00	20	60.0	7470.0	.00
21	60.0	7595.0	.00	22	60.0	7595.0	.00
23	60.0	7595.0	.00	24	60.0	7594.0	.00
25	60.0	7593.0	.00	26	60.0	7593.0	.00
27	95.0	7595.0	.00	28	60.0	7365.0	.00
29	60.0	7365.0	.00	30	60.0	7290.0	.00
31	57.0	7289.0	.00	32	47.0	7284.0	.00
33	95.0	7690.0	.00	34	95.0	7591.0	.00
35	55.0	7289.0	.00				

NETWORK CALCULATION RESULTS

AIRWAY	FROM	TO	TYPE	RESISTANCE	AIRFLOW	HEADLOSS
1	5	32	0	.1959E+01	5528.	.006
2	5	6	0	.4778E+01	42000.	.843
3	8	30	0	.5783E+01	10269.	.061
4	6	35	0	.4555E-03	14899.	.000
1050	35	8	0	.9110E+00	14899.	.020

5	6	7	0	.1318E+00	27101.	.010
6	7	31	0	.1318E+00	16997.	.004
7	30	9	0	.5727E+00	10269.	.006
8	8	10	0	.1237E+02	4629.	.027
9	7	12	0	.2405E+01	10104.	.025
10	31	14	0	.1002E+00	16997.	.003
11	9	28	0	.1250E+01	17759.	.039
12	10	9	0	.6769E+01	7489.	.038
13	11	10	0	.1779E+01	2860.	.001
14	12	11	0	.7908E+00	10045.	.008
15	13	12	0	.1711E+00	-60.	.000
16	14	13	0	.7791E+00	16000.	.020
17	14	29	0	.4648E+01	997.	.000
18	28	15	0	.2780E+00	17759.	.009
19	11	16	0	.1687E+02	7185.	.087
20	13	19	0	.8123E+00	16059.	.021
21	29	20	0	.4555E+03	997.	.045
22	16	15	0	.2002E+01	696.	.000
23	16	17	0	.9510E+00	6489.	.004
24	18	17	0	.9960E+00	2458.	.001
25	19	18	0	.6766E+01	10712.	.078
26	19	20	0	.4626E+01	3234.	.005
27	15	22	0	.1137E+01	18454.	.039
28	17	23	0	.3384E+01	8947.	.027
29	18	24	0	.4496E+01	8254.	.031
30	19	26	0	.2877E+03	2113.	.128
31	20	26	0	.6916E+02	4231.	.124
32	21	27	0	.3943E+01	19957.	.157
33	22	21	0	.1406E+01	21537.	.065
34	23	22	0	.8224E+01	3083.	.008
35	23	24	0	.1972E+01	3846.	.003
36	24	25	0	.8451E+00	12100.	.012
37	25	26	0	.1191E+01	8179.	.008
38	26	2	0	.4980E+01	14523.	.105
39	21	1	0	.4555E+03	1580.	.114
40	25	1	0	.1112E+03	3921.	.171
41	33	27	0	.7212E-06	7520.	.000
42	27	34	0	.3037E-07	27477.	.000
43	34	3	0	.1146E-05	42000.	.000
44	4	3	0	.1184E-05	5528.	.000
46	32	4	0	.5000E+03	5528.	1.528
47	23	1	0	.4555E+03	2018.	.186
48	1	33	0	.6073E-05	7520.	.000
49	2	34	0	.5000E+01	14523.	.105

FANS

AIRWAY	FROM	TO	AIRFLOW	FAN PRESSURE
45	3	5	47528.	1.444

THESE CHARACTERISTICS WERE STORED FOR FANS 45

Q*1000	PF	Q*1000	PF	Q*1000	PF	Q*1000	PF	Q*1000	PF
37.5	2.00	41.5	1.83	45.0	1.64	48.5	1.44	52.5	1.17
55.0	.98	57.5	.78	60.0	.58	62.5	.36	65.0	.15

THE STATED NUMBER OF AIRWAYS WAS 50
THE STATED NUMBER OF JUNCTIONS WAS 35

DATA FOR CONCENTRATION AND TEMPERATURE CALCULATION

AIRWAY	FROM	TO	ELEV. DIFF.	ROCK T	CH4 PROD.	COND.	DIFFUSIVITY
1	5	32	-2.00	60.00	.0	3.00	.100
2	5	6	3.00	60.00	.0	3.00	.100
3	8	30	1.00	60.00	.0	3.00	.100
4	6	35	.00	60.00	.0	3.00	.100
1050	35	8	.00	60.00	.0	3.00	.100
5	6	7	.00	60.00	.0	3.00	.100
6	7	31	.00	60.00	.0	3.00	.100
7	30	9	75.00	60.00	.0	3.00	.100
8	8	10	76.00	60.00	.0	3.00	.100
9	7	12	76.00	60.00	.0	3.00	.100
10	31	14	76.00	60.00	.0	3.00	.100
11	9	28	.00	60.00	.0	3.00	.100
12	10	9	.00	60.00	.0	3.00	.100
13	11	10	.00	60.00	.0	3.00	.100
14	12	11	.00	60.00	.0	3.00	.100
15	13	12	.00	60.00	.0	3.00	.100
16	14	13	.00	60.00	.0	3.00	.100
17	14	29	.00	60.00	.0	3.00	.100
18	28	15	105.00	60.00	.0	3.00	.100
19	11	16	105.00	60.00	.0	3.00	.100
20	13	19	105.00	60.00	.0	3.00	.100
21	29	20	105.00	60.00	.0	3.00	.100
22	16	15	.00	60.00	.0	3.00	.100
23	16	17	.00	60.00	.0	3.00	.100
24	18	17	.00	60.00	.0	3.00	.100
25	19	18	.00	60.00	.0	3.00	.100
26	19	20	.00	60.00	.0	3.00	.100
27	15	22	125.00	60.00	.0	3.00	.100
28	17	23	125.00	60.00	.0	3.00	.100

29	18	24	124.00	60.00	.0	3.00	.100
30	19	26	123.00	60.00	.0	3.00	.100
31	20	26	123.00	60.00	.0	3.00	.100
32	21	27	.00	60.00	.0	3.00	.100
33	22	21	.00	60.00	.0	3.00	.100
34	23	22	.00	60.00	.0	3.00	.100
35	23	24	-1.00	60.00	.0	3.00	.100
36	24	25	-1.00	60.00	.0	3.00	.100
37	25	26	.00	60.00	.0	3.00	.100
38	26	2	-2.00	60.00	.0	3.00	.100
39	21	1	95.00	60.00	.0	3.00	.100
40	25	1	97.00	60.00	.0	3.00	.100
41	33	27	-95.00	94.69	.0	3.00	.100
42	27	34	-4.00	94.71	.0	3.00	.100
43	34	3	-151.00	94.53	.0	3.00	.100
44	4	3	156.00	95.43	.0	3.00	.100
45	3	5	-154.00	60.00	.0	3.00	.100
46	32	4	.00	60.00	.0	3.00	.100
47	23	1	95.00	60.00	.0	3.00	.100
48	1	33	.00	95.00	.0	3.00	.100
49	2	34	.00	95.00	.0	3.00	.100

A TEMPERATURE OF 70.0 WAS ASSIGNED TO JUNCTION NO. 3

THE FOLLOWING CONTAMINATION WAS ASSUMED

	FIXED INFLUX			O2-RICH	FUEL-RICH		LEADING T	REF. Q
AIRWAY	FLUX	%CONC	HEAT	%O2	FUMES	HEAT		
4	200.0	100.00	.500E+05	.00	.00	.0	.00	0

TIME AT 30.0 HOUR(S) AFTER EVENT

THRESHOLD IN ACCURACY (SUM OF NVP CORRECTIONS PER MESH < 2.E-4 IN.W.G.)
SATISFIED. CURRENT SUMFNV PER MESH .000084 IN.W.G., ITERATIONS 4

TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN AIRWAYS

AIRWAY	FROM	TO	AIRFLOW	AVE. T	T AT END	FUMES	METHANE	HEADLOSS
1	5	32	5648.	65.80	63.04	00.00	00.00	.006
2	5	6	43284.	67.16	65.01	00.00	00.00	.869
3	8	30	9713.	118.03	84.35	1.01	00.00	.064
4	6	35	19519.	208.61	208.61	1.01	00.00	.000
1050	35	8	19519.	190.47	173.87	1.01	00.00	.051
5	6	7	23765.	64.88	64.75	00.00	00.00	.007
6	7	31	14698.	64.59	64.42	00.00	00.00	.003
7	30	9	9713.	81.95	79.69	1.01	00.00	.006
8	8	10	9805.	159.62	146.59	1.01	00.00	.160
9	7	12	9067.	64.14	63.56	00.00	00.00	.019
10	31	14	14698.	63.92	63.43	00.00	00.00	.002
11	9	28	18483.	70.11	68.74	1.01	00.00	.042
12	10	9	8769.	84.15	62.68	1.01	00.00	.054
13	10	11	1036.	108.90	84.15	1.01	00.00	.000
14	12	11	6588.	62.60	61.82	00.00	00.00	.003
15	12	13	2479.	63.26	62.98	00.00	00.00	.000
16	14	13	13835.	63.25	63.08	00.00	00.00	.014
17	14	29	863.	61.21	60.25	00.00	00.00	.000
18	28	15	18483.	67.32	66.04	1.01	00.00	.009
19	11	16	7625.	63.68	62.67	.14	00.00	.094
20	13	19	16314.	62.52	62.00	00.00	00.00	.021
21	29	20	863.	59.91	59.72	00.00	00.00	.032
22	16	15	1744.	61.02	60.25	.14	00.00	.001
23	16	17	5880.	62.50	62.33	.14	00.00	.003
24	18	17	2881.	60.76	60.64	00.00	00.00	.001
25	19	18	10840.	61.38	60.91	00.00	00.00	.076
26	19	20	3351.	60.85	60.26	00.00	00.00	.005
27	15	22	20227.	64.62	63.77	.94	00.00	.045
28	17	23	8761.	60.98	60.35	.09	00.00	.025
29	18	24	7959.	60.42	60.01	00.00	00.00	.027
30	19	26	2124.	60.44	59.75	00.00	00.00	.123
31	20	26	4213.	59.85	59.65	00.00	00.00	.116
32	21	27	21410.	62.69	62.34	.84	00.00	.173
33	22	21	22798.	63.21	63.07	.84	00.00	.070
34	23	22	2571.	60.15	60.04	.09	00.00	.005
35	23	24	4287.	60.24	60.15	.09	00.00	.003
36	24	25	12246.	60.06	60.05	.03	00.00	.012
37	25	26	8593.	60.04	60.03	.03	00.00	.008
38	26	2	14930.	59.92	59.95	.02	00.00	.105
39	21	1	1388.	60.85	59.90	.84	00.00	.083
40	25	1	3653.	59.84	59.67	.03	00.00	.140
41	33	27	6944.	94.92	94.96	00.00	00.00	.000
42	27	34	28354.	95.00	95.00	00.00	00.00	.000
43	34	3	43284.	94.97	95.02	00.00	00.00	.000
44	4	3	5648.	95.14	95.01	00.00	00.00	.000
45	3	5	48932.	69.93	69.86	00.00	00.00	1.427
46	32	4	5648.	63.00	62.95	00.00	00.00	1.524
47	23	1	1903.	59.97	59.75	.09	00.00	.156
48	1	33	6944.	95.00	95.00	00.00	00.00	.000
49	2	34	14930.	92.79	95.00	.02	00.00	.119

DATA FOR FIRE SOURCES

AIRWAY	AIRFLOW	TEMPERATURE	FUME	O2 LEFT	HEAT INPUT
4	19519.	208.63	1.01	21.00	.500E+05

PARAMETERS OF AIR IN JUNCTIONS

JUNCTION	TEMP.	FUMES	METHANE	JUNCTION	TEMP.	FUMES	METHANE
1	95.00	00.00	00.00	2	59.95	.02	00.00
3	70.00	00.00	00.00	4	95.00	00.00	00.00
5	69.86	00.00	00.00	6	65.01	00.00	00.00
7	64.75	00.00	00.00	8	173.87	1.01	00.00
9	71.63	1.01	00.00	10	146.59	1.01	00.00
11	64.85	.14	00.00	12	63.56	00.00	00.00
13	63.06	00.00	00.00	14	63.43	00.00	00.00
15	65.54	.94	00.00	16	62.67	.14	00.00
17	61.78	.09	00.00	18	60.91	00.00	00.00
19	62.00	00.00	00.00	20	60.15	00.00	00.00
21	63.07	.84	00.00	22	63.35	.84	00.00
23	60.35	.09	00.00	24	60.06	.03	00.00
25	60.05	.03	00.00	26	59.89	.02	00.00
27	95.00	00.00	00.00	28	68.74	1.01	00.00
29	60.25	00.00	00.00	30	84.35	1.01	00.00
31	64.42	00.00	00.00	32	63.04	00.00	00.00
33	95.00	00.00	00.00	34	95.00	00.00	00.00
35	208.61	1.01	00.00				

REVERSAL OF AIRFLOW HAS OCCURRED IN THE FOLLOWING PLACES

AIRWAY	13 IS NOW CARRYING AIR FROM	10 TO	11
AIRWAY	15 IS NOW CARRYING AIR FROM	12 TO	13

IN THE FOLLOWING AIRWAYS EXIST CRITICAL CONDITIONS

AIRWAY	FROM	TO	CH4 % > .050	FUMES % > .000	TEMPERATURE > 95. F	HEADLOSS < .010 IN.WG.
1	5	32	00.00	00.000	63.0	.006
3	8	30	00.00	1.014	84.3	.064
4	6	35	00.00	1.014	208.6	.000
1050	35	8	00.00	1.014	173.9	.051
5	6	7	00.00	00.000	64.8	.007
6	7	31	00.00	00.000	64.4	.003
7	30	9	00.00	1.014	79.7	.006

8	8	10	00.00	1.014	146.6	.160
10	31	14	00.00	00.000	63.4	.002
11	9	28	00.00	1.014	68.7	.042
12	10	9	00.00	1.014	62.7	.054
13	10	11	00.00	1.014	84.1	.000
14	12	11	00.00	00.000	61.8	.003
15	12	13	00.00	00.000	63.0	.000
17	14	29	00.00	00.000	60.2	.000
18	28	15	00.00	1.014	66.0	.009
19	11	16	00.00	.137	62.7	.094
22	16	15	00.00	.137	60.3	.001
23	16	17	00.00	.137	62.3	.003
24	18	17	00.00	00.000	60.6	.001
26	19	20	00.00	00.000	60.3	.005
27	15	22	00.00	.938	63.8	.045
28	17	23	00.00	.092	60.3	.025
32	21	27	00.00	.843	62.3	.173
33	22	21	00.00	.843	63.1	.070
34	23	22	00.00	.092	60.0	.005
35	23	24	00.00	.092	60.2	.003
36	24	25	00.00	.032	60.1	.012
37	25	26	00.00	.032	60.0	.008
38	26	2	00.00	.019	60.0	.105
39	21	1	00.00	.843	59.9	.083
40	25	1	00.00	.032	59.7	.140
41	33	27	00.00	00.000	95.0	.000
42	27	34	00.00	00.000	95.0	.000
43	34	3	00.00	00.000	95.0	.000
44	4	3	00.00	00.000	95.0	.000
47	23	1	00.00	.092	59.8	.156
48	1	33	00.00	00.000	95.0	.000
49	2	34	00.00	.019	95.0	.119

IN THE FOLLOWING JUNCTIONS EXIST CRITICAL CONDITIONS

JUNCTION	CH4 % > .05	FUMES % > .000	TEMP. F > 95.	JUNCTION	CH4 % > .05	FUMES % > .000	TEMP. F > 95.
1	00.00	00.00	95.0	2	00.00	.02	60.0
4	00.00	00.00	95.0	8	00.00	1.01	173.9
9	00.00	1.01	71.6	10	00.00	1.01	146.6
11	00.00	.14	64.9	15	00.00	.94	65.5
16	00.00	.14	62.7	17	00.00	.09	61.8
21	00.00	.84	63.1	22	00.00	.84	63.4
23	00.00	.09	60.3	24	00.00	.03	60.1
25	00.00	.03	60.1	26	00.00	.02	59.9
27	00.00	00.00	95.0	28	00.00	1.01	68.7
30	00.00	1.01	84.3	33	00.00	00.00	95.0
34	00.00	00.00	95.0	35	00.00	1.01	208.6

☆U.S. Government Printing Office: 1990-711-297

INT.BU.OF MINES,PGH.,PA 29112

270 90

THE LIBRARY OF CONGRESS

THE LIBRARY OF CONGRESS

HECKMAN
BINDERY INC.
JUN 90
N. MANCHESTER,
INDIANA 46962

LIBRARY OF CONGRESS
0 002 951 131 7